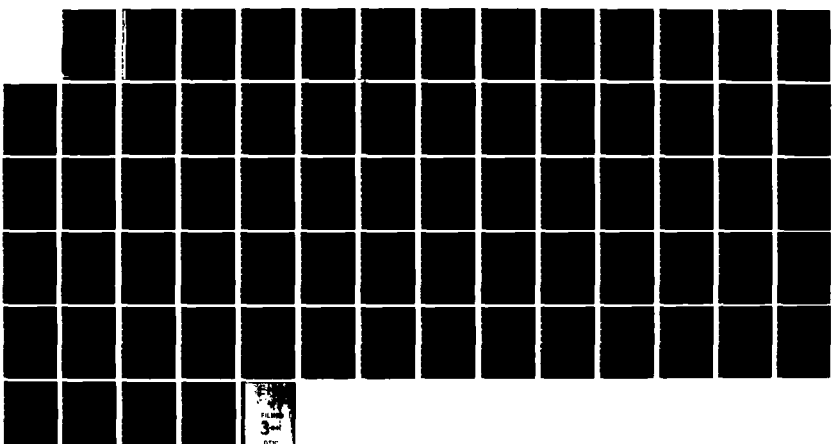
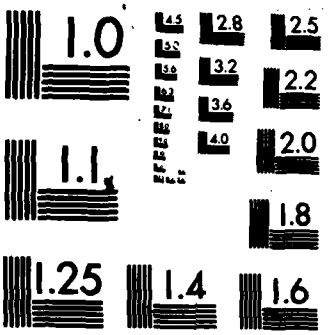
AD-A137 554 AN OVERVIEW OF RECURSIVE LEAST SQUARES ESTIMATION AND LATTICE FILTERS(U) STANFORD UNIV CA INFORMATION SYSTEMS LAB J M TURNER JAN 84 ISL-TR-M736-2 N00014-82-K-0492 1/1

UNCLASSIFIED F/G 12/1 NL





(2)

INFORMATION SYSTEMS LABORATORY

STANFORD ELECTRONICS LABORATORIES
DEPARTMENT OF ELECTRICAL ENGINEERING
STANFORD UNIVERSITY · STANFORD, CA 94305

AN OVERVIEW OF RECURSIVE LEAST SQUARES ESTIMATION AND LATTICE FILTERS

by

John M. Turner

Technical Report No. M736-2

Partially Supported by

Office of Naval Research, Contract N00014-82-K-0492,

Defense Advanced Research Project Agency,

Contract MDA903-82-K-0382 (DSN).

January 1984

AN OVERVIEW OF RECURSIVE LEAST SQUARES ESTIMATION AND LATTICE FILTERS

by

John M. Turner

Technical Report No. M736-2

Partially Supported by

Office of Naval Research, Contract N00014-82-K-0492,

Defense Advanced Research Project Agency,

Contract MDA903-82-K-0382 (DSN).

January 1984

TABLE OF CONTENTS

AN OVERVIEW OF

RECURSIVE LEAST SQUARES ESTIMATION

AND LATTICE FILTERS †

John M. Turner

1. INTRODUCTION

The lattice filter structure is an alternative means of realizing a digital filter transfer functions. Although the lattice filter structure (also called the ladder structure) does not have the minimum number of multipliers and adders for a transfer function realization, it does have several advantageous properties. These include cascading of identical sections, coefficients with magnitudes less than one, stability test by inspection, and good numerical roundoff characteristics. Moreover, the lattice filter structure is particularly suited for adaptive filtering since the recursive solution of least squares estimators naturally produces a lattice filter structure. Also the lattice filter structure orthogonalizes the input signal on a stage by stage basis. This leads to very fast convergence and tracking capabilities of the lattice structure. Although many alternative techniques have been developed to estimate the reflection coefficients that parameterize the lattice structure, the recursive least squares method updates the least squares estimate upon the observation *of* each data sample. This procedure leads to an optimal estimate and requires only slightly greater computational burden than alternative techniques.

This overview presents the derivation of the recursive least squares lattice filter using a recursive extension of the standard block data Levinson least squares solution. The linear predic-

† This work was partially supported by the Office of Naval Research under Contract N00014-82-K-0402 and partially used facilities supported by the Defense Advanced Research Projects Agency under Contract MDA903-82-K-0032 (DSN).

to be published in *Adaptive Filters*, Cowan and Grant (Editors), Prentice Hall, 1984.

tor filter presented here has been widely used for synthesis of speech waveforms (LPC), deconvolution of seismic data, high resolution spectral estimation, adaptive line enhancement, adaptive noise cancelling and adaptive antenna array processing. The ideas embodied in this estimation technique are derived from the work of many individuals since 1970. The approach presented here follows that of [Lee, 1980].

Adaptive estimation techniques modify the estimation filter parameters according to the newly observed data sample. For every new data sample, recursive estimation using the lattice filter generates new reflection coefficients and prediction errors for every filter order. Changing every filter coefficient for each new data sample is important for applications where fast convergence or tracking of quickly time varying signals is required. However, for applications where the dynamics are slow, only the results after observing the signal for a certain time period are important. The recursive algorithms described here can also be used to accumulate signal properties over a particular time period. The procedure for converting between lattice filter coefficients and the more common equivalent tapped delay line filter coefficients is given in Section 2.

The mathematics of recursive least squares estimation requires the updating of variables with time and order subscripts. The algorithms for this are often complicated. Therefore, an intuitive introduction in Section 3 presents the nature of the lattice filter structure, the stage by stage orthogonalizing property, and analogies with physical phenomena. Section 4 briefly presents approximation techniques for determining the reflection coefficients from observed data. The advantage of the lattice filter structure is that *time recursive exact least squares solutions* to estimation problems can be efficiently computed. The development of the Recursive Least Squares Lattice estimation algorithm is presented in Section 5 and 6. A square root normalized least squares lattice algorithm that has better numerical properties is presented in Section 7.

The computational complexity of these algorithms is discussed in Section 8. An efficient means of implementing the recursive least squares algorithm using rotational arithmetic is presented. This rotational arithmetic, called CORDIC arithmetic, is not new, having been used for calculating trigonometric functions in hand held calculators. The design of an integrated cir-

cuit chip to implement the least squares lattice algorithm using CORDIC arithmetic is mentioned.

To demonstrate the power of this adaptive estimator, simulation examples of convergence and tracking, examples using real speech data and electrophysiological data and adaptive equalizer examples are presented in Section 9. Since recursive least squares estimation and lattice filter structures have been a very active area of research, Section 10 refers to related ideas.

Since the equations developed here are recursive in order and time, the following notation is used. A variable $x(t)$ is a general time sampled data value while x_T is the specific data sample T samples after the beginning of the recursion (relative time T). Bold capital letter variables represent matrices or vectors. When two subscripts are used, the first is the order and the second is the time parameter, ie. $\mathbf{A}_{i,T}$ is the vector of i-th order predictor coefficients determined from data up to the specific time T.

2. GENERAL LATTICE DIGITAL FILTER STRUCTURE

The general lattice digital filter is a means of realizing a digital filter transfer function. The lattice structure itself is motivated by similar analog filter structures that have good properties. In this section, the digital lattice filter is introduced and related to the direct form tapped delay line digital filter.

Since analog lattice and ladder filters have desirable characteristics, digital filters with similar structures were investigated. For example, third order LC Butterworth filter shown in Fig. 1a is a simple analog lattice-type structure. It is noted for the relative insensitivity of its frequency response to slight perturbations of the circuit element values around their nominal values. This structure can be transformed into the general lattice network shown in Fig. 1b. The latter is the lattice structure of interest in this chapter. The analog lattice structure consists of a cascade of identical stages, each stage with a pair of input and output terminals. By developing a digital filter configuration that is similar to the analog lattice structure, the digital filter inherits many of the same properties. Since the structure of a digital filter realization influences its sensitivity to finite word length arithmetic, the digital lattice filter has good numerical properties.

The digital lattice filter realizations consist of cascaded stages with two input and two output ports, as in the analog structures. Possible digital lattice configurations for realizing a general digital transfer function include: an asymmetric multiplier form (Fig. 2) and a symmetric two multiplier form (Fig. 3). For the asymmetric multiplier lattice, the structure inside each stage realizes a single pole and zero equivalent transfer function. The algorithm for determining the asymmetric multiplier structure of Fig. 2 can be found in [Mitra et al, 1977]. This form degenerates to a tapped delay line for either all pole or all zero transfer functions and thus is not of interest here. The symmetric two multiplier form does not degenerate but it requires more multipliers than an equivalent tapped delay line filter. This lattice filter can be modified to a one multiplier form so as to have the minimum number of multipliers, but this requires extra adders (Fig. 3b).

A cascade of lattice sections, forming a lattice filter can implement a digital transfer function in a way that has advantages over the standard direct form, parallel, or standard cascade form realizations. The cascaded structure in the lattice filter propagates a forward signal $f_j(t)$ and a backward signal $b_j(t)$ at time t and section number j. The fundamental equation describing the lattice filter structure is (1), see Figure 4.

$$f_{j+1}(t) = f_j(t) - k_{j+1}\, b_j(t-1) \qquad (1)$$
$$b_{j+1}(t) = b_j(t-1) - k_{j+1}\, f_j(t)$$

The multipliers in the crossover portion of the lattice, k_i are known as reflection coefficients or partial correlation (PARCOR) coefficients.

The implementation of digital filter transfer functions in lattice form have been examined [Gray and Markel, 1973,1975]. State space canonical forms were also established [Morf, 1974, Morf et al, 1977, Lee, 1980]. ALGORITHM 1 determines the reflection coefficients k_i and tap coefficients v_i for the lattice filter of Fig. 3 that is equivalent to a (stable nonreducible) direct form transfer function with numerator coefficients b_i^p and denominator coefficients a_i^p (from [Gray and Markel, 1973]). The one multiplier form in Fig 3b uses coefficients v_i^o. Although the lattice coefficients and the direct form coefficients are related in a nonlinear manner, this algorithm is invertible so that a lattice structure can be converted uniquely to a direct form filter and vice versa (when all the roots are inside the unit circle).

ALGORITHM 1:

General Transfer Function to Lattice Filter

$$H_P(z) = \frac{\sum_{j=0}^{P} b_j^P \; z^{-j}}{\sum_{j=0}^{P} a_j^P \; z^{-j}} \quad ; \quad a_0^P = 1$$

$s_P = 1$
For $i = P$ to 1
  $k_i = -a_i^i$
  $v_i = b_i^i$
  $v_i^e = v_i \; / \; s_i$
  $s_{i-1} = s_i \; (\; 1 + k_i \;)$
  For $j = 1$ to $i-1$

$$a_j^{i-1} = \frac{a_j^i + k_i \; a_{i-j}^i}{1 - k_i^2}$$

$$b_j^{i-1} = b_j^i - v_i \; a_{i-j}^i$$

  continue
  $b_0^{i-1} = b_0^i + v_i \; k_i$
continue
$v_0 = b_0^0$
$v_0^e = v_0 \; / \; s_i$
END

While Fig. 3 and ALGORITHM 1 describe the lattice filter for a general transfer function with poles and zeros, the remainder of this chapter discusses all pole transfer functions and their inverses, all zero transfer functions. For an all pole transfer function ($b_0^P = 1$, $b_j^P = 0$, $j > 1$), the lattice filter is called the feedback lattice filter shown in Fig. 4. The inverse of the feedback lattice can be determined by applying Mason's rule to Fig. 4. This finite impulse response filter, an all zero transfer function, is the feedforward lattice (Fig.5). Thus a feedforward lattice and a feedback lattice with the same coefficients perform inverse operations on the input signal. If a signal is applied to a feedforward lattice filter and the result is applied to a feedback lattice filter, the original signal is returned. According to Mason's rule, the reflection coefficients parameterize both the feedback and feedforward lattice with the appropriate change in signal flow. ALGORITHM 2 gives the procedure for converting from reflection coefficients to tapped delay line coefficients where the signal flow specifies whether the all pole or all zero transfer function is implied.

ALGORITHM 2:

Lattice Coefficients to Tapped Delay Line Coefficients

$a_1^1 = -k_1$
For $i = 2$ to P
    $a_i^i = -k_i$
    For $j = 1$ to $i-1$
        $a_j^i = a_j^{i-1} - k_i\ a_{i-j}^{i-1}$
    continue
continue
END

The coefficient sensitivity of the lattice implementation of a general digital transfer function has not been studied as thoroughly as other common filter structures. Scaling conventions [Markel and Gray, 1975(2)] and roundoff noise characteristics [Markel and Gray, 1975(1)] for finite wordlength arithmetic were developed for various lattice configurations. The one multiplier lattice has roundoff noise characteristics that are always better than the two multiplier lattice. Both lattice filters are always better than the direct form realization. Particularly when the width of the filter pass band becomes small, the The lattice filter structure has better roundoff noise characteristics than other filter realizations, particularly when the width of the filter pass band becomes small. A normalized lattice filter (requiring more multiplications) was developed that performs better than the other lattice structures or parallel form realizations.

The implementation of a transfer function requires quantized coefficients that can effect the stability of a filter and its inverse. The sensitivity of the roots of the transfer function to perturbations of the lattice filter and tapped delay line filter coefficients has been investigated [Chu and Messerschmitt, 1980,1983]. The effect of varying the tapped delay line filter coefficients was the same for each coefficient. When the roots are close to the unit circle, quantization of the tapped delay line coefficients tends to move the roots perpendicular to the unit circle. For an all pole transfer function, this quantization can cause the poles to move outside the unit circle and the transfer function to become unstable. For lattice filters, the effect on the root location of varying the low order coefficients is much greater than for the higher order coefficients. For reflection coefficients, the roots tend to move tangentially to the unit circle, thus changing the center

frequency rather than the bandwidth of the roots. Low order reflection coefficients, particularly those with magnitudes near unity need to be more accurately quantized than the higher order reflection coefficients. No simple rule of thumb exists for tapped delay line filters. The effects of the lattice coefficient quantization have been studied most extensively for speech modeling applications. For typical prediction filters used in speech processing, substantially coarser quantization of the reflection coefficients than of the tapped delay line filter coefficients is possible while still maintaining the subjectively perceived spectral response.

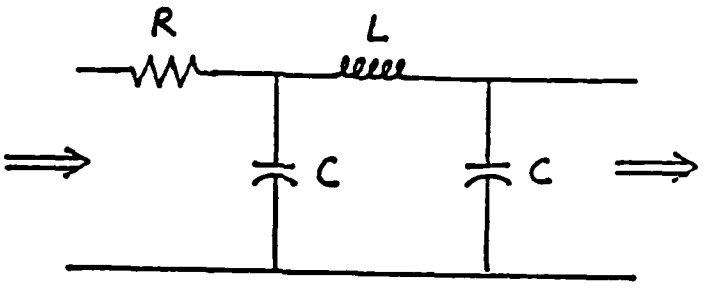


Fig. 1a Third Order LC analog Ladder

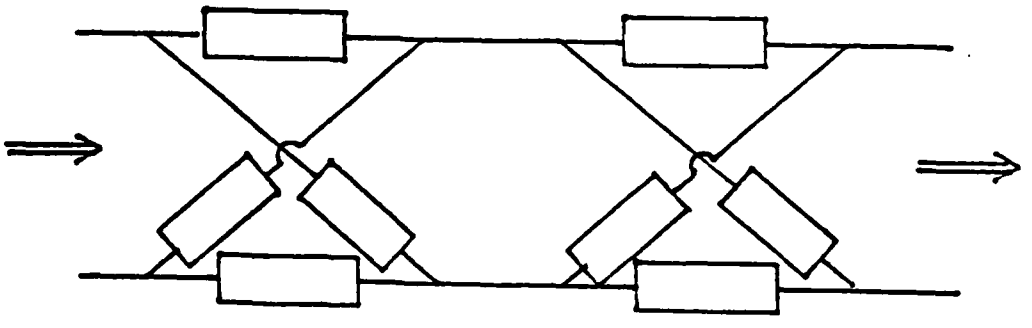


Fig. 1b General Analog Impedance Lattice

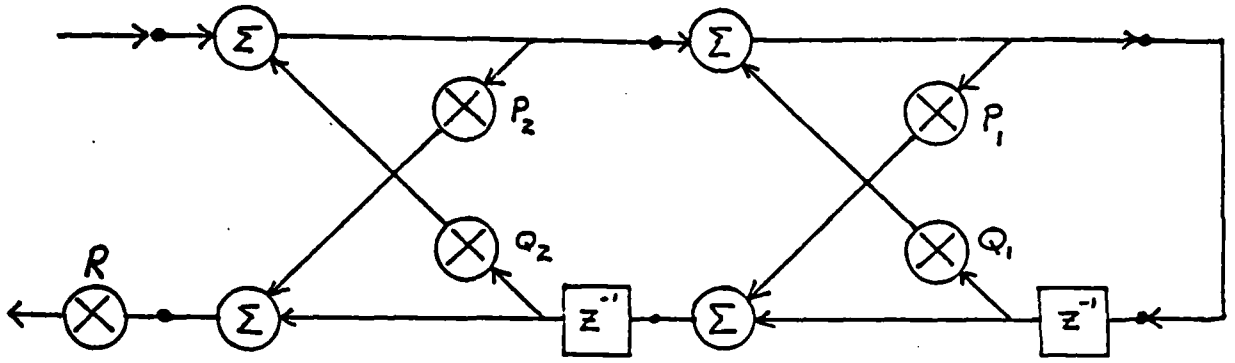


Fig. 2 Digital Asymmetric Lattice- Pole and Zero Transfer Function

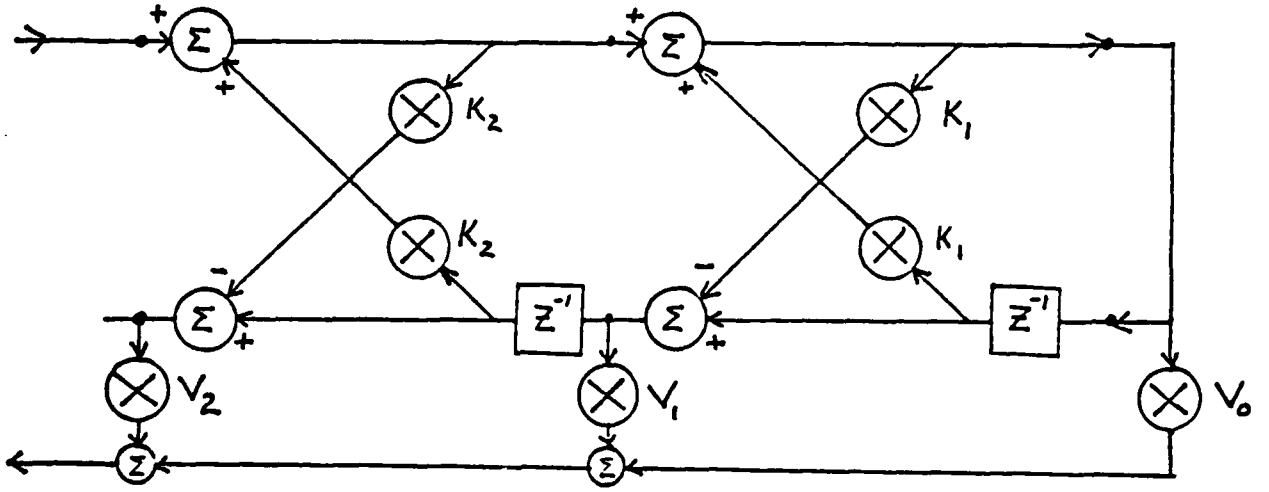


Fig. 3a Digital Symmetric Two Multiplier Lattice - Pole and Zero Transfer Function

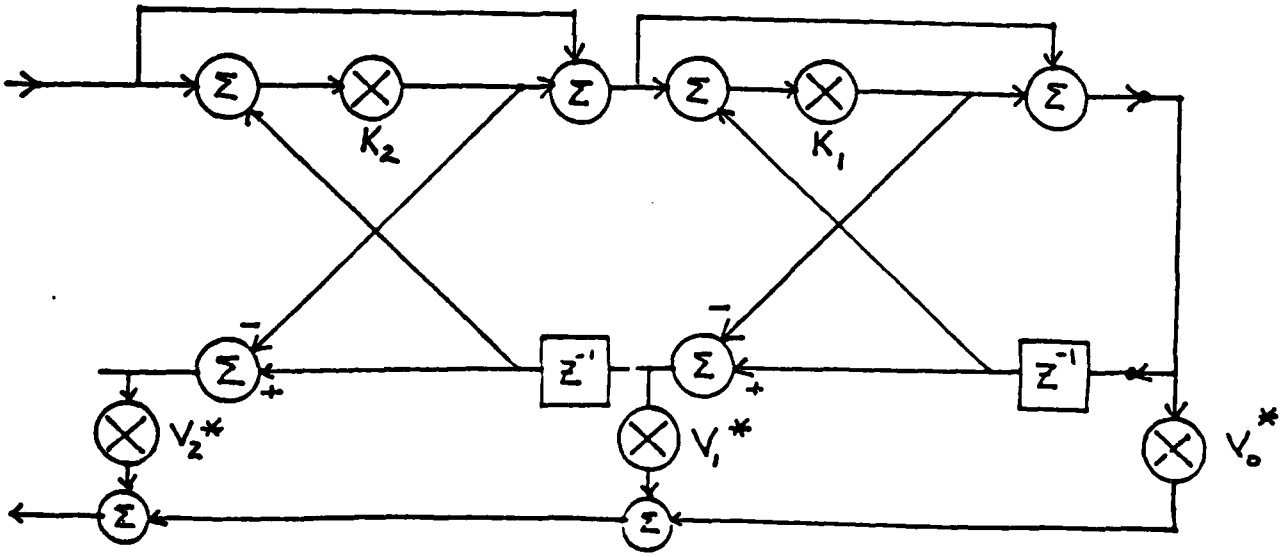


Fig. 3b Digital Symmetric One Multiplier Lattice - Pole and Zero Transfer Function

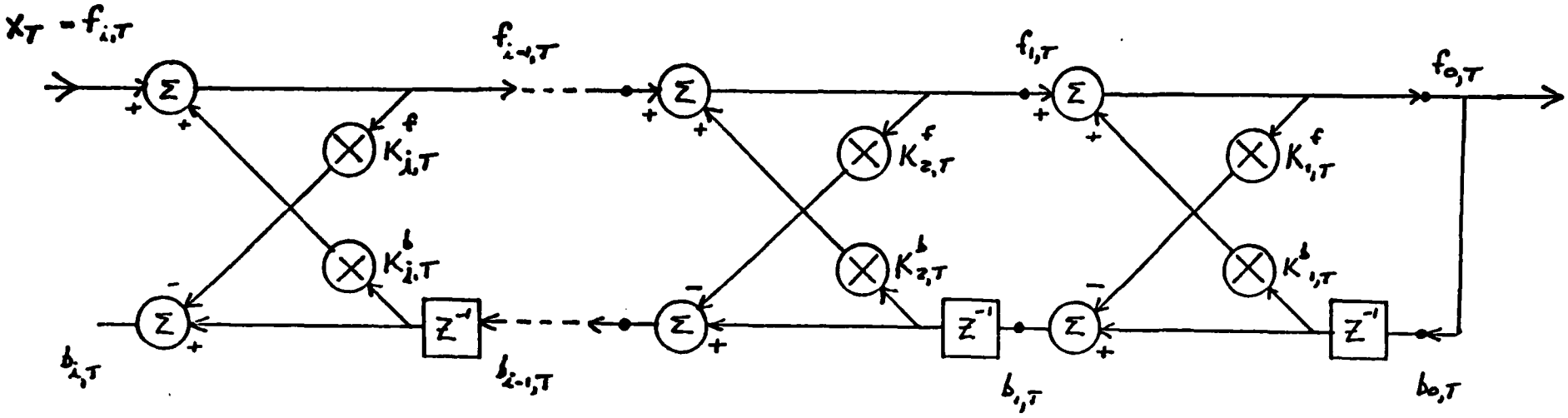


Fig. 4 Feedback Lattice Filter - All Pole Transfer Function

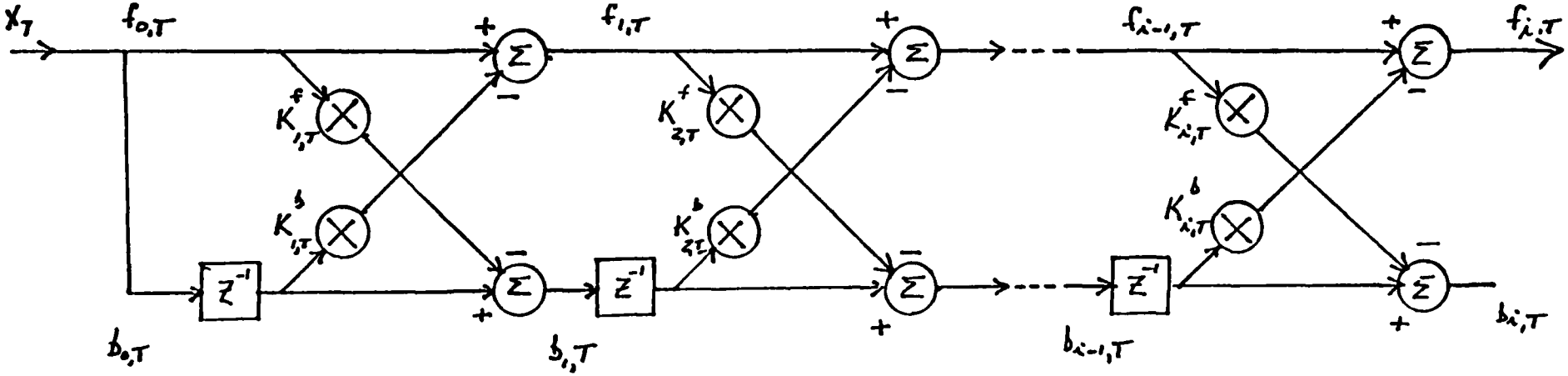


Fig. 5 Feedforward Lattice Filter - All Zero Transfer Function

3. PROPERTIES OF THE LATTICE STRUCTURE

The lattice filter has a more complex structure and requires more numerical operations to implement a transfer function than the straight forward direct form realization. However this increased complexity is offset by several advantageous properties of the lattice structure including; a stability test by inspection, a stage by stage orthogonalization of the input signal, and a physical interpretation as wave propagation in a stratified medium. The lattice filter structure naturally evolves from a prediction filter where orthogonality conditions are applied. The properties of the lattice estimation filter are presented in this section. An acoustical tube model of the human vocal tract is interpreted as a lattice filter in Section 3.2 to lend a physical interpretation to the reflection coefficients.

3.1 Orthogonalising Properties

In early investigations of the lattice structure, a connection with orthogonal polynomials was noted [Itakura and Saito, 1968, Burg, 1975, Makhoul, 1975, Markel and Gray, 1976]. Lattice form realizations are obtained by an orthogonalization of the state-space transfer function using Szego orthogonal polynomials. The theory of Szego polynomials and their applications in system theory (stability testing) and in stochastic problems (prediction theory and spectral analysis) has been discussed in [Grenander and Szego]. The Schur stability test uses the properties of orthogonal polynomials to determine whether the poles of a transfer function are inside the unit circle and hence a stable transfer function. The test is performed by using ALGORITHM 1 to compute the lattice filter then checking that the magnitudes of all the reflection coefficients $\{k_i\}$ are less than one.

For problems in estimation, the minimum mean squared error estimation equations can be transformed into a stage by stage optimization by forming a recursion on the optimum filter order. The parameter in the recursion can be estimated stage by stage since it depends on quantities that are orthogonal between stages. This orthogonalization property is now developed for the

prediction filter (an all zero transfer function).

For a predictor of p-th order, the data sample at time t, $x(t)$, is approximated as a linear combination of the previous p samples, $x(t-1),...,x(t-p)$. The forward prediction error $f_p(t)$ must be orthogonal to the previous data samples to attain the minimum mean squared error value. This determines the weighting factors, $\{a_i\}$ on the previous data.

$$f_p(t) = x(t) + a_1 x(t-1) + \cdots + a_p x(t-p) \tag{2}$$
$$E(f_p(t)\, x(t-j)) = 0, \quad 1 \le j \le p$$

The operation $E(\cdot)$ represents the statistical expectation. This prediction error is also called a prediction residual or an innovation when the coefficients are chosen to attain the minimum mean squared error. A backward prediction error, $b_p(t-1)$ can similarly be defined to predict $x(t-p-1)$ from the same samples, $x(t-1),...,x(t-p)$.

$$b_p(t-1) = x(t-p-1) + c_1 x(t-p) + \cdots + c_p x(t-1) \tag{3}$$
$$E(b_p(t-1)\, x(t-j)) = 0, \quad 1 \le j \le p$$

Here the $\{c_j\}$ are chosen to satisfy this condition. Notice that both prediction errors satisfy the same orthogonality conditions.

Increasing the prediction order to $p+1$, $f_{p+1}(t)$ represents the component of $x(t)$ that is not predictable from $x(t-1),...,x(t-p),x(t-p-1)$. The p-th prediction error uses information up to $x(t-p)$, so now the information about $x(t)$ that can be predicted from $x(t-p-1)$ must be included. However much of this information is already contained in $x(t-1),...,x(t-p)$. The backward prediction error $b_p(t-1)$ represents the new information in the sample $x(t-p-1)$. The plausible recursion for $f_{p+1}(t)$ is (4) where the scalar k^f_{p+1} is determined so that $f_{p+1}(t)$ satisfies the new orthogonality conditions.

$$f_{p+1}(t) = f_p(t) - k^f_{p+1}\, b_p(t-1) \tag{4}$$
$$E(f_{p+1}(t)\, x(t-j)) = 0, \quad 1 \le j \le p+1$$

The only constraint not immediately satisfied involves $x(t-p-1)$, and is given by (5). By substi-

tuting (3) and (4) in (5), the optimal k^f_{p+1} is determined (6).

$$E(f_{p+1}(t)\, x(t-p-1)) = 0 \tag{5}$$
$$0 = E(f_p(t)\, x(t-p-1)) - k^f_{p+1}\, E(b_p(t-1)\, x(t-p-1))$$
$$= E(f_p(t)\, b_p(t-1)) - k^f_{p+1}\, E(b^2_p(t-1))$$

$$k^f_{p+1} = E(f_p(t)\, b_p(t-1)) \,/\, E(b^2_p(t-1)) \tag{6}$$

Similarly, the recursion for the backward predictor is obtained (7) and the optimal k^b_{p+1} is determined (8).

$$b_{p+1}(t) = b_p(t-1) - k^b_{p+1}\, f_p(t) \tag{7}$$
$$E(b_{p+1}(t-1)\, x(t)) = 0$$

$$0 = E(f_p(t)\, b_p(t-1)) - k^b_{p+1}\, E(f^2_p(t))$$
$$k^b_{p+1} = E(f_p(t)\, b_p(t-1)) \,/\, E(f^2_p(t)) \tag{8}$$

Extending the prediction filter to the next higher order, $p+2$ requires the calculation of the new prediction errors, f_{p+1} and b_{p+1} from (4) and (7). Thus a prediction filter can be constructed solely using the lattice structure by successively increasing the filter order. This is the stage by stage orthogonalization property of the lattice structure where each reflection coefficient is determined separately. This stage by stage computation of prediction coefficients does not hold for the tapped delay line filter (2). The coefficients $\{a_j\}$ are interdependent and they all change when the filter order increases.

Further insight into properties of the prediction errors is provided in [Makhoul, 1978(2)]. The backward prediction error results from a Gram-Schmidt type orthogonalization of delayed versions of the signal. This property of orthogonal variables makes the lattice structure advantageous for adaptive filtering. Also the decrease in signal energy after each prediction stage is easily determined. This feature can be used to scale the prediction errors to maintain good numerical properties. The most important properties are summarized here.

$$E(f_p(t)\, b_j(t-1)) = \begin{cases} \Delta & j = p \\ 0 & 1 \le j < p \end{cases} \tag{9}$$

$$E(f_p^2(t)) = E(f_p(t)x(t)) = \sigma_p^f \qquad (10)$$
$$E(f_p(t)\, f_j(t)) = \sigma_p^f \quad 1 \le j \le p$$

$$E(b_p^2(t-1)) = E(b_p(t-1)x(t-p-1)) = \sigma_p^b \qquad (11)$$
$$E(b_p(t-1)\, b_j(t-1)) = \begin{cases} \sigma_p^b & j = p \\ 0 & 1 \le j < p \end{cases}$$

$$\sigma_{p+1}^f = \sigma_p^f \left(1 - k_p^f\, k_p^b \right) \qquad (12)$$
$$\sigma_{p+1}^b = \sigma_p^b \left(1 - k_p^f\, k_p^b \right)$$

When the signal $x(\cdot)$ is stationary with known autocorrelation function, the forward and backward prediction error energies at each stage are identical ($\sigma_p^f = \sigma_p^b$). Then the two reflection coefficients are equal and the symmetric two multiplier lattice structure computes these prediction error recursions. When the signal to be modeled is assumed to be stationary, a single reflection coefficient, k is determined by combining sample data estimates of k^f and k^b. This lattice filter with constant coefficients is the feedforward lattice (1) of Section 2. For nonstationary signals, adaptive estimates are generated by making the reflection coefficients time varying.

The reflection coefficients are closely related to partial correlation factors which have several interesting statistical properties. The correlation between $x(t)$ and $x(t-p-1)$, after their mutual linear dependence on the intervening samples $\{x(t-1), \ldots, x(t-p)\}$ has been removed is $E(f_p(t)\, b_p(t))$. This relation arises from the orthogonalizing nature of the lattice. When this correlation is normalized by the variance of f_p and b_p, it is known as the p-th order partial correlation. The autocorrelation function of a stationary unit variance discrete time process can be uniquely characterized by a sequence of reflection coefficients, having values less than or equal to one [Barndorff-Nielsen and Schou, Ramsey]. For any p-th order AR process, the partial correlation of higher order, lag $p + i$, $(i = 1, 2, \ldots)$ is zero. For a stationary AR process, the sample estimates of the partial correlations are asymptotically Gaussian and independent (see [Murthy and Narasimham] for more statistical properties).

In applications such as noise cancelling or equalization, the orthogonalizing properties of the lattice are of primary interest to obtain fast tracking or convergence, see Section 6. The back-

ward prediction errors, $b_j(t)$ are extensively used since they are a Gram-Schmidt orthogonalization of delayed versions of the input time series.

3.2 Physical Interpretation

The lattice structure and the reflection coefficients have a physical interpretation that for particular classes of signals lends understanding to the properties of the lattice structure. Modeling of wave propagation in a stratified medium leads to a cascade of lattice filters. This model has been applied in seismic signal processing by [Treitel and Robinson, Burg, 1967] and others. The physical properties of scattering medium leads to inversion methods based on cascaded reflection elements, eg. the characterization of (electrical) transmission lines [Gopinath and Sondhi, 1971] or the human vocal tract [Gopinath and Sondhi, 1970]. Similarly in the fields of acoustics and speech processing, an acoustic transmission line with step changes in impedance leads to a lattice cascade structure. The human vocal tract has been modeled as a cascade of acoustic tube sections with different impedances. This relationship between a physiological system and the lattice structure gives a physical meaning to the reflection coefficients and led to the development of speech synthesis systems using the lattice structure. The remainder of this section develops an acoustical tube model of the vocal tract into a lattice filter (see [Flanagan, Markel and Gray, 1976, Rabiner and Schafer, 1978]).

A lossless acoustical tube transmission line composed of cascaded cylinders of differing diameter but equal length was developed as a model of the vocal tract in [Kelly and Lochbaum]. This vocal tract model was studied to obtain a better understanding of the speech production mechanism and to synthesize speech by computer. Speech sounds result from pressure waves resonating in the vocal tract (acoustic tube). The significance of the model is that the cascaded cylinders become cascaded lattice stages. The cross sectional areas of adjacent cylinders specify reflected and transmitted acoustic wave components which translate into the lattice reflection coefficients.

Sound waves that propagate in a cylindrical section obey the conservation of momentum and mass equations (assuming standard conditions, see [Rabiner and Schafer, 1978]). Since the

cross-sectional area of the n-th tube is constant, combining the conservation laws yields a one dimensional wave equation. To satisfy this equation, the steady state volume velocity $u_i(z,t)$ and the pressure wave $p_i(z,t)$ are composed of waves traveling in the forward, u^+ and backward, u^- direction.

$$\frac{\partial p}{\partial z} = -\frac{\rho}{A}\frac{\partial u}{\partial t}$$
$$\frac{\partial u}{\partial z} = -\frac{A}{\rho c^2}\frac{\partial p}{\partial t}$$
$$u_i(z,t) = u_i^+(t-z/c) - u_i^-(t+z/c) \tag{13}$$
$$p_i(z,t) = \rho c/A_i \ (u_i^+(t-z/c) + u_i^-(t+z/c))$$

The density of air is ρ, c is the speed of sound in air and A_i is the cross-sectional area of the acoustic tube, see Fig. 6a. Assuming that all the tubes are of equal length, L, at the boundary between tubes i and $i+1$, a continuous wave propagation is required.

$$u_i(L,t) = u_{i+1}(0,t)$$
$$p_i(L,t) = p_{i+1}(0,t)$$

Using the boundary conditions, the transmitted wave u_{i+1}^+ and the reflected wave u_i^- across the boundary are determined.

$$u_{i+1}^+(t) = (1 + k_i) u_i^+(t-\tau) + k_i u_{i+1}^-(t) \tag{14}$$
$$u_i^-(t+\tau) = -k_i u_i^+(t-\tau) + (1 - k_i) u_{i+1}^-(t)$$

Here $\tau = L/c$ is the propagation time through the tube section and k_i is the wave reflection coefficient at the junction of A_i and A_{i+1}.

$$k_i = (A_{i+1} - A_i) / (A_{i+1} + A_i) \tag{15}$$

Since the cross-sectional areas are all positive, $-1 \le k_i \le 1$. The wave propagation due to the discontinuity in cross-sectional area, is shown in Fig. 6a,b.

The lattice filter structure is obtained by normalizing variables and grouping time delays. By modifying (14), the waves in the i-th physical section at the boundary with the $(i+1)$-th section can be written in terms of the $(i+1)$-th section.

$$u_i^+(t-\tau) = (u_{i+1}^+(t) - k_i u_{i+1}^-(t)) / (1 + k_i) \tag{16}$$
$$u_i^-(t+\tau) = (u_{i+1}^-(t) - k_i u_{i+1}^+(t)) / (1 + k_i)$$

An absolute time reference is established at the output of the last tube section, which physically

would be at the lips. Assuming that the vocal tract model has p tube sections, the time delay from the beginning of the i-th tube to the lips is $t_i = (\, p + 1 - i)\, \tau$, so the time variable in the equation for the i-th section is replaced by $t - t_i$. A scale factor c_i is introduced to combine the $(\, 1 + k_j)$ factors from the i-th tube to the last (p-th) section (lips).

$$c_i = \prod_{j=i}^{p} (\, 1 + k_j\,) = (\, 1 + k_i\,)\, c_{i+1}$$

The lattice equations are obtained from (16) by using the absolute time reference and defining new variables, see Fig. 6c.

$$\begin{aligned} f_i(t) &= c_i\ u_i^{+}(t - \tau - t_i) \\ b_i(t) &= c_i\ u_i^{-}(t + \tau - t_i) \\ f_i(t) &= f_{i+1}(t) - k_i\ b_{i+1}(t - 2\tau) \\ b_i(t) &= b_{i+1}(t - 2\tau) - k_i\ f_{i+1}(t) \end{aligned} \tag{17}$$

This is the same equation as that developed earlier, (4) and (7), from orthogonality conditions except the unit delay is 2τ and the lattice sections are numbered in decreasing order.

Although modeling of the entire vocal tract includes other influences due to the vocal chords (glottis) and lip radiation, the wave propagation in the mouth ideally follows the lattice structure equations. Studies have indicated that every reasonable vocal tract shape could be generated by a lattice filter and that the reflection coefficients are directly related to the cross sectional area of the vocal tract [Markel and Gray, 1976].

For other types of signals, if they are generated by or can be modeled as wave propagation in a stratified medium, the lattice structure is intuitively motivated. For physically generated processes, the process is often nonstationary but there is a limit to the rate at which a process can change. The shape of the vocal tract (excluding the lips) can only change at a moderately slow rate determined primarily by the muscles in the tongue. Except when a sudden opening of the lips occurs, the cross-sectional area of the vocal tract changes slowly and hence the reflection coefficients also change slowly. This slow time evolution can be used advantageously in adaptive estimation or parameter quantization.

An intuitive understanding of the significance of reflection coefficient values is possible

because the lattice filter structure can be thought of as wave propagation in the acoustical tube. The equivalent tapped delay line coefficients are not as easily interpreted. While the reflection coefficients are limited to $-1 \leq k_i \leq 1$, the equivalent tapped delay line coefficients are often a factor of ten times larger. When the reflection coefficient is zero, the signal propagates without change since adjoining sections in the tube would have the same cross sectional area. When the reflection coefficient is near -1, the signal is apt to have highly resonant or oscillator characteristics since if the next tube section is completely closed, ie. zero cross section, then the wave is totally reflected. Conversely, when the reflection coefficient is near + 1, a decaying signal amplitude is usually found since if the cross sectional area increases greatly across a boundary, then there is full forward radiation. This connection between the physical properties of wave propagation in the acoustical tube and the analogous lattice filter structure greatly aids in an intuitive understanding of the effect of reflection coefficient values on signal characteristics.

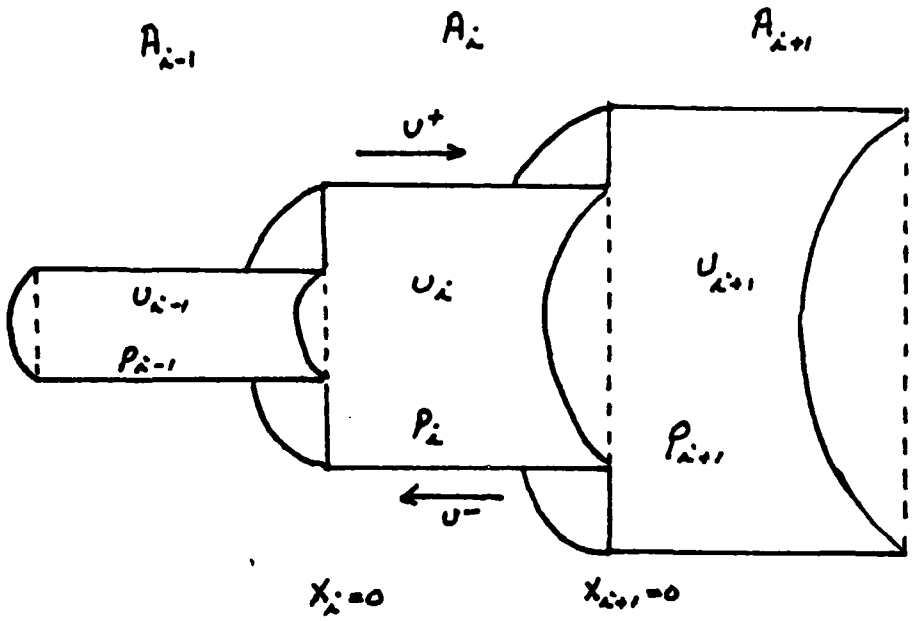


Fig. 5.6a Acoustic Tube

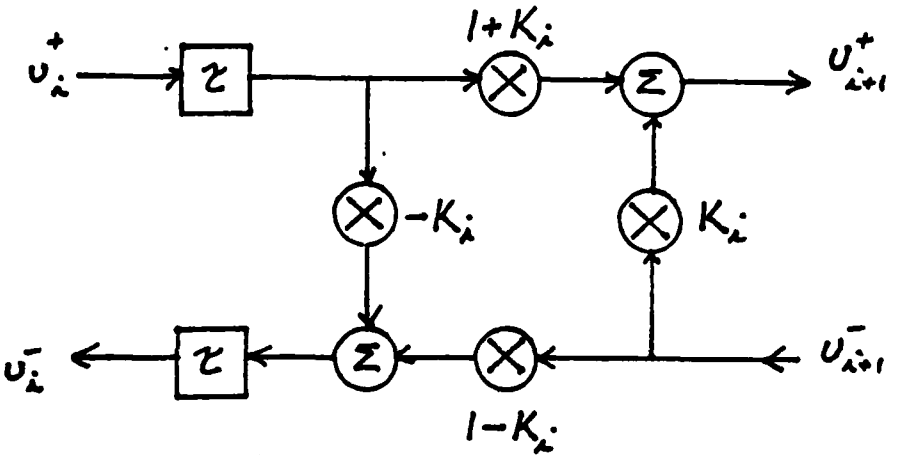


Fig. 5.6b Discontinuity: Tube i to Tube $i+1$

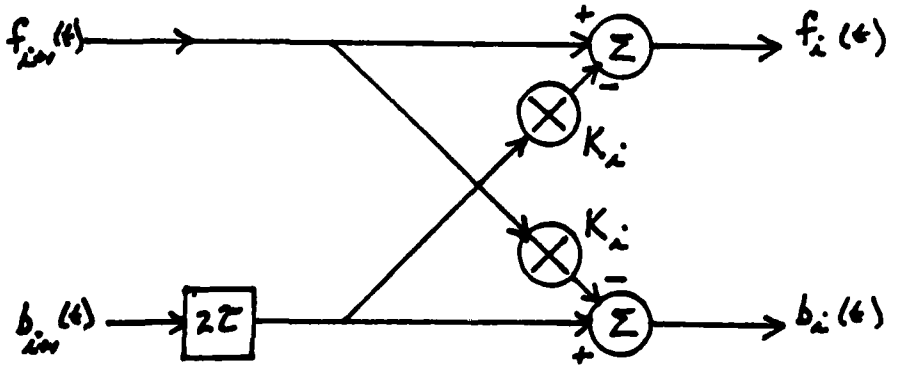


Fig. 5.6c Equivalent Lattice Filter

Fig. 6 Acoustic Tube Interpreted as a Lattice Filter

4. SAMPLE DATA ESTIMATES OF REFLECTION COEFFICIENTS

The real advantage of the lattice filter is for adaptive filtering, where the characteristics of an unknown process are to be determined from the observed data samples. The remainder of this overview presents the adaptive lattice filter where each new data sample is used to update the reflection coefficients. In this section, approximation techniques that estimate the reflection coefficients based on gradient approaches or sample (block) data estimates of statistical quantities are presented. The development of a recursive exact solution for least squares estimation, which naturally produces a lattice structure filter, begins in Section 5. A simpler set of recursive equations using normalized variables is presented in Section 7.

The earliest techniques for estimating reflection coefficients assumed that the signal was locally stationary. Therefore sample data approximations were used for the statistical definition of the reflection coefficients, (6) and (8). When the process $x(\cdot)$ is stationary with known autocorrelation function, the forward and backward prediction error energies at each stage are identical, $(\sigma_j^f = \sigma_j^b)$. Thus the reflection coefficient for the forward and backward predictor are the same and the lattice filter stage requires a single parameter.

$$f_{j+1}(t) = f_j(t) - k_{j+1}\, b_j(t-1) \qquad (18)$$
$$b_{j+1}(t) = b_j(t-1) - k_{j+1}\, f_j(t)$$

The block data techniques use a time sequence of data and determine a single prediction filter for this entire block of data. A single reflection coefficient per lattice stage is calculated by combining sample data estimates of k^f and k^b. If the geometric mean of k^f and k^b is used, then the reflection coefficient becomes the correlation coefficient between f_j and b_j. This parameter, k^f was originally called a partial correlation (PARCOR) coefficient [Itakura and Saito, 1968]. It is the normalized conditional correlation coefficient between $x(t)$ and $x(t-j-1)$ given the intervening data samples, $x(t-1),\ldots,x(t-j)$.

$$k_{j+1}^f = \frac{\sum_{t=1}^{T} f_j(t)\, b_j(t-1)}{\sqrt{\sum_{t=1}^{T} f_j^2(t) \cdot \sum_{t=1}^{T} b_j^2(t-1)}} \qquad (19)$$

The expression for k_{j+1} that minimizes $E(f^2_{j+1}(t)) + E(b^2_{j+1}(t))$ is the harmonic mean of k^f and k^b. This estimate, k^B is computationally simpler and is related to Burg's maximum entropy method [Burg, 1975].

$$k^B_{j+1} = \frac{\sum_{t=1}^{T} f_j(t)\, b_j(t-1)}{1/2 \sum_{t=1}^{T} (f^2_j(t) + b^2_j(t-1))} \tag{20}$$

These two definitions for reflection coefficients use blocks of T data samples and thus require many computations. For k_{j+1}, the T samples of f_j and b_j are required. To calculate k_{j+2}, the lattice filtering at stage j+1 must be performed to obtain T new values for f_{j+1} and b_{j+1}. Clearly this requires P filtering steps of 2T samples to determine P reflection coefficients.

Adaptive gradient algorithms for determining the reflection coefficients greatly reduce the computational complexity of the estimation technique. Only the prediction errors at the previous time instant is needed for the gradient methods. The block data approach required all the past error values. Several techniques have been proposed for adapting the reflection coefficients for every newly observed data sample. These techniques do not minimize any criterion but try to change the reflection coefficient in the direction of decreasing prediction error energy (gradient descent). Two classes of gradient techniques either approximate the reflection definition of (20) as the current reflection coefficient plus a correction or approximate the numerator and denominator separately. The simplest update of the reflection coefficient [Griffiths, 1977] uses the forward and backward prediction errors weighted by the constant α.

$$k_j(t+1) = k_j(t) + \alpha \{ f_j(t)\, b_{j-1}(t-1) + f_{j-1}(t)\, b_j(t) \} \tag{21}$$

This estimate can be improved by replacing the weighting factor by an energy normalized term, $1 / \sigma_j(t)$ where $\sigma_j(t)$ is the accumulated average of $f^2_{j-1}(t)$ and $b^2_{j-1}(t-1)$ [Griffiths, 1978, Makhoul, 1978(2)].

$$k_j(t+1) = k_j(t) + (1 / \sigma_j) \{ f_j(t)\, b_{j-1}(t-1) + f_{j-1}(t)\, b_j(t) \} \tag{22}$$
$$\sigma_j(t) = (1-\beta)\, \sigma_j(t-1) + f^2_{j-1}(t) + b^2_{j-1}(t-1)$$

Another adaptive estimate [Makhoul and Viswanathan, 1978(1), Makhoul, 1978(2)] approximates the numerator and denominator of (20) separately. The same weighting factor α is used for

both terms. The ratio of these two terms becomes the estimate of the reflection coefficient.

$$c_j(t+1) = (1-\alpha)\, c_j(t) + 2\, f_{j-1}(t)\, b_{j-1}(t-1)$$
$$d_j(t+1) = (1-\alpha)\, d_j(t) + f_{j-1}^2(t) + b_{j-1}^2(t-1)$$

$$k_j(t+1) = \frac{c_j(t+1)}{d_j(t+1)} \tag{23}$$

This ratio is a biased estimator since in general $E(x/y) \neq E(x)/E(y)$ but simulations indicate that this bias is generally very small [Honig and Messerschmitt, 1981].

The convergence of the lattice filter is much faster than that of the adaptive tapped delay line filter [Satorius and Alexander, 1979(2), Horvath]. This is because the lattice filter tries to orthogonalize the input signal so that the coefficient estimates are decoupled. In fact, the convergence time is almost independent of the eigenvalue spread of the signal, i.e. independent of the signal's spectral dynamic range [Griffiths, 1977]. Quantitative characterizations of the convergence properties of the gradient reflection coefficient estimators (22) and (23) have been studied [Honig and Messerschmitt, 1981]. A two stage gradient lattice algorithm was compared with a two stage LMS gradient tapped delay line filter to show that it is possible but unlikely for the tapped delay line filter to converge faster than the lattice filter. A comparison of lattice estimation techniques using the gradient and block data reflection coefficient definitions (21), (22), and (23) has been presented in [Gibson and Haykin].

This orthogonalizing and decoupling property of the lattice is only asymptotically obtained using the gradient estimation techniques of this section. The recursive least squares lattice, developed in the next section, exactly solves the orthogonalization for every new data sample. The optimal solution is similar to the energy normalized gradient lattice (22) except that the optimum weighting factor is computed (instead of the constant β). This least squares lattice estimation technique has even faster convergence than the gradient lattice methods of above. However as the number of data samples (from a stationary process) gets large, the results from the gradient lattice and the least squares lattice become similar.

5. RECURSIVE LEAST SQUARES LATTICE ALGORITHM

The recursive Least Squares Lattice algorithm (LSL) allows the exact solution to the least squares problem to be updated for every newly observed data sample. This adaptive estimation technique uses the properties of the lattice filter to efficiently implement the adaptation. The LSL algorithm looks similar to the gradient techniques of the previous section except that optimal weighting factor are calculated. The LSL algorithm is developed in this section in the context of an extension to the Levinson algorithm for solving the normal equation.

The least squares solution to a linear modeling problem can be reduced to a simple set of linear equations called the normal or Yule-Walker equations. These equations, which involve the inversion of a covariance matrix, has been widely studied to reduce the computational burden, guarantee stable models, and handle nonstationary processes. The linear predictor form of the linear modeling problem is presented here.

The linear prediction model assumes that a data sample at time T, x_T can be approximated as $\hat{x}_T$, a weighted sum of previous data samples. For an $p-th$ order linear predictor with coefficients $(a_1, \ldots, a_p)$:

$$\hat{x}_{p,T} = - a_1 x_{T-1} - \cdots - a_p x_{T-p} \tag{24}$$

The coefficients are to be chosen so as to minimize the mean squared error between x_T and the estimate, $\hat{x}_{p,T}$. The $p-th$ order covariance matrix of the process $x(\cdot)$ is $\mathbf{R}_p$ and is composed of elements $r_{i,j}$.

$$\begin{aligned} \mathbf{R}_p &= \mathbf{E}\left[\mathbf{x}_{|T:T-p|}\, \mathbf{x}^{T}_{|T:T-p|}\right] \\ \mathbf{x}_{|T:T-p|} &= [x_T,\ x_{T-1}, \ldots, x_{T-p}]^T \\ r_{i,j} &= E[x_{T-i}\, x_{T-j}],\quad 0 \le i,j \le p \end{aligned} \tag{25}$$

Minimizing the square of the prediction error with respect to the predictor coefficients, $\{a_i\}$ requires that the predictor coefficients satisfy (26), called the normal equation, where σ_p is the

minimum error.

$$\min_{\{a_i\}} E\{(x_T - \hat{x}_{p,T})^2\} \quad \rightarrow \quad \mathbf{R}_p \begin{bmatrix} 1 \\ a_1 \\ . \\ . \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma_p \\ 0 \\ . \\ . \\ 0 \end{bmatrix} \tag{26}$$

Unless the process is a deterministic one, a unique solution to (26) exists.

In general, for a p-th order linear model, the solution of the normal equation involves the inversion of the p by p covariance matrix. Standard matrix inversion methods such as Gaussian elimination require $O(p^3)$ computations (multiplications). However for stationary random processes, the covariance matrix is a Toeplitz matrix.

$$r_{i,j} = r_{|i-j|}, \quad 0 \leq i,j \leq p$$

Using the Levinson algorithm, the normal equation in Toeplitz form can be solved in $O(p^2)$ computations. The Levinson algorithm is an order recursive technique that uses the solution for an i–th order predictor to generate the solution for the $(i+1)$–th order predictor. This algorithm performs an orthogonalization as discussed is Section 3. The reflection coefficients are related to the predictor coefficients and are generated as a byproduct of this algorithm. In the Toeplitz case, the reflection coefficients can be determined directly without using predictor coefficients [LeRoux and Gueguen]. This is an application of the Schur algorithm. If the covariance sequence $(r_0, r_1, \ldots, r_p)$ is fed into a growing order lattice filter, the reflection coefficient at each stage can be determined by dividing the forward error by backward error at the input of that stage [Morf et al, 1977].

5.1 Formulation of Recursive Estimates

In the development of recursive least squares lattice algorithms, two aspects of the solution of the normal equation are important. The first aspect is the efficient inversion of the covariance matrix, Toeplitz and non-Toeplitz, that gives rise to the order-update recursions. Secondly, the time-update structure allows exact least squares solutions to be computed in a recursive manner for each new data sample. This enables the lattice algorithms to achieve extremely fast

convergence and excellent tracking capability. The first derivation of the least squares lattice came from extending the Levinson approach and was based on the predictor coefficients. Once the recursions were obtained, it was noted that they could be written compactly using just the lattice filter parameters. A subsequent direct derivation of the recursive equation, using a geometric approach solely in terms of reflection coefficients and lattice parameters was reported in [Lee, 1980, Lee et al, 1981, Shensa, 1981]. An overview of the first derivation is presented here since it gives insight into the nature of the recursions. The development of the order- and time-update recursions for the lattice algorithms based on the special structures of the normal equation follows the approach in [Lee, 1980].

First the structural properties of the sample covariance matrix must be exploited as the order of the predictor changes and as new time samples are added. The covariance matrix of order p for data samples $\{x_j,\ j=0,T\}$ used here is the prewindowed form.

$$\mathbf{R}_{i,T} = \mathbf{X}_{i,T}\,\mathbf{X}_{i,T}^{T} \tag{27}$$

$$\mathbf{X}_{i,T} = \begin{bmatrix} x_0 & \cdots & x_i & \cdots & x_T \\ & \ddots & & & \vdots \\ & & \ddots & & \vdots \\ 0 & & x_0 & \cdots & x_{T-i} \end{bmatrix} \tag{28}$$

The covariance matrix $\mathbf{R}_{i,T}$ has the following structural properties. As a new data sample x_{T+1} is included, the new covariance matrix is composed of the previous covariance matrix plus a matrix of special form. The time-update matrix equation is given in (29).

$$\mathbf{R}_{i,T+1} = \mathbf{R}_{i,T} + \begin{bmatrix} x_{T+1} \\ \vdots \\ x_{T-i+1} \end{bmatrix} \left[x_{T+1}, \ldots, x_{T-i+1} \right] \tag{29}$$

Also, the covariance matrix of order $i+1$ contains the covariance matrix of order i. The order-update matrix equation for the covariance are given in (30) where $*$ denotes unspecified elements along the outer row and column.

$$\mathbf{R}_{i+1,T} = \begin{bmatrix} * & * \\ * & \mathbf{R}_{i,T-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{i,T} & * \\ * & * \end{bmatrix} \tag{30}$$

The development of a recursive solution requires that the normal equation, (26) be extended with auxiliary vectors. The forward and backward predictors vectors are $\mathbf{A}_{i,T}$ and $\mathbf{C}_{i,T}$ respectively. A vector $\mathbf{D}_{i,T}$ is defined to account for new data samples. The matrix equation (31), an extension of (26) defines these vectors.

$$\mathbf{R}_{i,T}\,[\mathbf{A}_{i,T},\ \mathbf{C}_{i,T},\ \mathbf{D}_{i,T}] = \begin{bmatrix} \sigma^{f}_{i,T} & 0 & x_T \\ 0 & 0 & x_{T-1} \\ . & . & . \\ 0 & 0 & x_{T-i+1} \\ 0 & \sigma^{b}_{i,T} & x_{T-i} \end{bmatrix} \tag{31}$$

$$\mathbf{A}_{i,T} = [\,1,\ a^{i}_{1,T},\ \ldots,\ a^{i}_{i,T}\,]^{T}$$

$$\mathbf{C}_{i,T} = [\,c^{i}_{i,T},\ \ldots,\ c^{i}_{1,T},\ 1\,]^{T}$$

The forward prediction error, $f_{i,T}$ and the backward prediction error, $b_{i,T}$, are defined as in (2) and (3) with $\mathbf{x}_{|T:T-i|}$ defined as in (25).

$$f_{i,T} = \mathbf{A}^{T}_{i,T}\,x_{|T:T-i|} \tag{32}$$

$$b_{i,T} = \mathbf{C}^{T}_{i,T}\,x_{|T:T-i|}$$

To account for the end of the sample data set, an auxiliary vector $\mathbf{D}_{i,T}$ and related scalar $\gamma_{i,T}$ are introduced.

$$\mathbf{D}_{i,T} = R^{-1}_{i,T}\,x_{|T:T-i|} \tag{33}$$

$$\gamma_{i,T} = \mathbf{D}^{T}_{i,T}\,x_{|T:T-i|} = \mathbf{x}^{T}_{|T:T-i|}\,R^{-1}_{i,T}\,x_{|T:T-i|} \tag{34}$$

This parameter $\gamma_{i,T}$ can be interpreted as a likelihood variable and is limited to the range $0 \leq \gamma_{i,T} \leq 1$, see Section 5.5.

5.2 Order Update Equations

As in the Levinson solution, an efficient means of determining the $n-th$ order solution is to develop recursive equations for updating the predictor order (at a fixed time). Following the usual development of recursive equations, assuming the predictor vectors $\mathbf{A}_{i,T}$ and $\mathbf{C}_{i,T}$ are known, the predictors of order $i+1$ are to be determined. Thus the vectors $\mathbf{A}_{i+1,T}$ and $\mathbf{C}_{i+1,T}$

must satisfy the following normal equation.

$$\mathbf{R}_{i+1,T}\,[\mathbf{A}_{i+1,T},\ \mathbf{C}_{i+1,T}] = \begin{bmatrix} \sigma^{f}_{i+1,T} & | & 0 \\ 0 & | & 0 \\ \cdot & | & \cdot \\ 0 & | & 0 \\ 0 & | & \sigma^{b}_{i+1,T} \end{bmatrix}$$

Since $\mathbf{A}_{i+1,T}$ is to be assembled from $\mathbf{A}_{i,T}$ and $\mathbf{C}_{i,T-1}$, these i-th order predictors are augmented with a final (or initial) zero to make an $i+1$ vector. From (30), the augmented $\mathbf{A}_{i,T}$ satisfies the new normal equation except for the last entry.

$$\mathbf{R}_{i+1,T}\begin{bmatrix}\mathbf{A}_{i,T}\\ 0\end{bmatrix} = \begin{bmatrix}\mathbf{R}_{i,T} & *\\ * & *\end{bmatrix}\begin{bmatrix}\mathbf{A}_{i,T}\\ 0\end{bmatrix} = [\,\sigma^{f}_{i,T},\ 0,\ \ldots,\ 0,\ \Delta_{i+1,T}]^{T} \tag{35}$$

$$\Delta_{i+1,T} = [\textit{last row of}\ \mathbf{R}_{i+1,T}]\begin{bmatrix}\mathbf{A}_{i,T}\\ 0\end{bmatrix} \tag{36}$$

Similarly, augmenting $\mathbf{C}_{i,T-1}$ with a leading zero satisfies the normal equations except for the first entry.

$$\mathbf{R}_{i+1,T}\begin{bmatrix}0\\ \mathbf{C}_{i,T-1}\end{bmatrix} = \begin{bmatrix}* & *\\ * & \mathbf{R}_{i,T-1}\end{bmatrix}\begin{bmatrix}0\\ \mathbf{C}_{i,T-1}\end{bmatrix} = [\Gamma_{i+1,T},\ 0,\ \ldots,\ 0,\ \sigma^{b}_{i,T-1}]^{T}\ , \tag{37}$$

$$\Gamma_{i+1,T} = [\textit{first row of}\ \mathbf{R}_{i+1,T}]\begin{bmatrix}0\\ \mathbf{C}_{i,T-1}\end{bmatrix}$$

Since $\mathbf{R}_{i+1,T}$ is symmetric, it can be seen that $\Delta_{i+1,T} = \Gamma_{i+1,T}$ [Bu75].

By appropriately combining these two equations to cancel the leading or final term in the normal equation, the order-update equations are obtained. Multiplying (37) by $\Delta_{i+1,T} / \sigma^{b}_{i,T-1}$ and then subtracting the result from (35), the order-update recursion for $\mathbf{A}_{i,T}$ and $\sigma^{f}_{i,T}$ are obtained.

$$\mathbf{A}_{i+1,T} = \begin{bmatrix}\mathbf{A}_{i,T}\\ 0\end{bmatrix} - \begin{bmatrix}0\\ \mathbf{C}_{i,T-1}\end{bmatrix}\Delta_{i+1,T} / \sigma^{b}_{i,T-1} \tag{38}$$

$$\sigma^{f}_{i+1,T} = \sigma^{f}_{i,T} - \Delta^{2}_{i+1,T} / \sigma^{b}_{i,T-1} \tag{39}$$

Similarly the order-update recursion for $\mathbf{C}_{i,T}$ and $\sigma^{b}_{i,T}$ are obtained.

$$\mathbf{C}_{i+1,T} = \begin{bmatrix} 0 \\ \mathbf{C}_{i,T-1} \end{bmatrix} - \begin{bmatrix} \mathbf{A}_{i,T} \\ 0 \end{bmatrix} \Delta_{i+1,T} / \sigma^{f}_{i,T} \tag{40}$$

$$\sigma^{b}_{i+1,T} = \sigma^{b}_{i,T-1} - \Delta^{2}_{i+1,T} / \sigma^{f}_{i,T} \tag{41}$$

When the order-update equations of the predictors are premultiplied by $[x_T, \ ..., \ x_{T-i-1}]$, the lattice equations result. The reflection coefficients are $k^{f}_{i+1,T}$ and $k^{b}_{i+1,T}$.

$$f_{i+1,T} = f_{i,T} - k^{b}_{i+1,T} \, b_{i,T-1} \tag{42}$$

$$b_{i+1,T} = b_{i,T-1} - k^{f}_{i+1,T} \, f_{i,T} \tag{43}$$

$$k^{f}_{i+1,T} = \Delta_{i+1,T} / \sigma^{f}_{i,T} \tag{44}$$

$$k^{b}_{i+1,T} = \Delta_{i+1,T} / \sigma^{b}_{i,T-1} \tag{45}$$

So far the development has been parallel to the non-adaptive approach in Section 3. To develop adaptive solutions, time-update equations for the predictors must be developed. Before proceeding to time-updates, the order-update recursion for $\mathbf{D}_{i,T}$ is developed in a similar fashion.

$$\mathbf{R}_{i+1,T} \begin{bmatrix} D_{i,T} \\ 0 \end{bmatrix} = \begin{bmatrix} R_{i,T} & * \\ * & * \end{bmatrix} \begin{bmatrix} D_{i,T} \\ 0 \end{bmatrix} = [x_T, \ x_{T-1}, \ ..., \ x_{T-i}, \ * \ ]^{T}$$

The last element of $\mathbf{D}_{i+1,T}$ can be found from the last row of $\mathbf{R}^{-1}_{i+1,T}$, by (32) and (33).

$$\textit{last element } \mathbf{D}_{i+1,T} = \textit{last element } (\mathbf{R}^{-1}_{i+1,T}) \, \mathbf{x}_{|T:T-i-1|} = b_{i+1,T} / \sigma^{b}_{i+1,T}$$

Since the last element of $\mathbf{C}_{i+1,T}$ is 1 and the last element of $\mathbf{D}_{i+1,T}$ has been determined, the order-update equation for $\mathbf{D}_{i+1,T}$ becomes (46).

$$\mathbf{D}_{i+1,T} = \begin{bmatrix} \mathbf{D}_{i,T} \\ 0 \end{bmatrix} + \mathbf{C}_{i+1,T} \, b_{i+1,T} / \sigma^{b}_{i+1,T} \tag{46}$$

The order-update for $\gamma_{i,T}$ is determined by premultiplying (46) by $[x_T, \ ..., \ x_{T-i-1}]$.

$$\gamma_{i+1,T} = \gamma_{i,T} + b^{2}_{i+1,T} / \sigma^{b}_{i+1,T} \quad . \tag{47}$$

5.3 Time Update Equations

Next the time-update of the covariance matrix is used to determine time-updates for the predictor vectors. From the time-update of the covariance matrix (29) and the definition of the

forward prediction error (32);

$$\mathbf{R}_{i,T+1}\,A_{i,T} = \begin{bmatrix} f \\ \sigma_{i,T} \\ 0 \\ . \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} z_{T+1} \\ z_T \\ . \\ z_{T-i} \\ z_{T-i+1} \end{bmatrix} [z_T\ ,\ \ldots\ ,\ z_{T-i}]^T \mathbf{A}_{i,T} \tag{48}$$

The auxiliary vector $\mathbf{D}_{i,T}$ is used to account for the new data samples.

$$\mathbf{R}_{i,T+1} \begin{bmatrix} 0 \\ \mathbf{D}_{i-1,T} \end{bmatrix} = \begin{bmatrix} * & * \\ * & \mathbf{R}_{i-1,T} \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{D}_{i-1,T} \end{bmatrix} = \begin{bmatrix} * \\ z_T \\ . \\ z_{T-i} \\ z_{T-i+1} \end{bmatrix} \tag{49}$$

The time-update recursion for $\mathbf{A}_{i,T}$ is obtained from (48) and (49).

$$\mathbf{A}_{i,T+1} = \mathbf{A}_{i,T} - \begin{bmatrix} 0 \\ D_{i-1,T} \end{bmatrix} [x_T\ ,\ \ldots\ ,\ x_{T-i}]^T \mathbf{A}_{i,T} \tag{50}$$

By premultiplying (50) by $[z_{T+1},\ \ldots,\ z_{T-i+1}]$ and using the definition of $\gamma_{i-1,T}$, (34) the expression can be simplified.

$$[z_T\ ,\ \ldots\ ,\ z_{T-i}]^T \mathbf{A}_{i,T} = f_{i,T+1}\ /\ (1 - \gamma_{i-1,T}) \tag{51}$$

The time-update for $\mathbf{A}_{i,T}$ becomes a simple expression.

$$\mathbf{A}_{i,T+1} = A_{i,T} - \begin{bmatrix} 0 \\ D_{i-1,T} \end{bmatrix} \frac{f_{i,T+1}}{1 - \gamma_{i-1,T}} \tag{52}$$

From the preceding relation, the time-update for $\sigma^f_{i,T}$ can be determined using (52) and (29).

$$\begin{aligned} \sigma^f_{i,T+1} &= \mathbf{A}^T_{i,T+1}\, R_{i,T+1}\, A_{i,T+1} \\ \sigma^f_{i,T+1} &= \sigma^f_{i,T} + \frac{f^2_{i,T+1}}{1 - \gamma_{i-1,T}} \end{aligned} \tag{53}$$

By applying similar techniques, the time-update recursions for $\mathbf{C}_{i,T}$ and $\sigma^b_{i,T}$ are determined.

$$\mathbf{C}_{i,T+1} = C_{i,T} - \begin{bmatrix} D_{i-1,T+1} \\ 0 \end{bmatrix} \frac{b_{i,T+1}}{1 - \gamma_{i-1,T+1}} \tag{54}$$

$$\sigma^b_{i,T+1} = \sigma^b_{i,T} + \frac{b^2_{i,T+1}}{1 - \gamma_{i-1,T+1}} \tag{55}$$

To recursively update the reflection coefficients, the time-update equation for Δ is needed.

$$[\,0\ \ \mathbf{C}^T_{i,T}\,]\, \mathbf{R}_{i+1,T+1} \begin{bmatrix} \mathbf{A}_{i,T+1} \\ 0 \end{bmatrix} = \Delta_{i+1,T+1} \tag{56}$$

By using the covariance time-update (29), the time-update for $\mathbf{C}_{i,T}$ and the forward and backward predictors, the time-update equation is obtained.

$$\Delta_{i+1,T+1} = \Delta_{i+1,T} + \frac{b_{i,T}\, f_{i,T+1}}{1 - \gamma_{i-1,T}} \tag{57}$$

The exact time-update of $\Delta_{i+1,T}$ is a time average of the cross-correlations between $b_{i,T}$ and $f_{i,T+1}$, with the special gain factor $\frac{1}{1 - \gamma_{i-1,T}}$. This relates $\Delta_{i+1,T}$ to the partial correlations discussed in the previous section.

The development of the time-update equations allows the influence of all prior data to be accumulated into the current parameter estimates. If the process that is being approximated contains time varying attributes, then it is desirable to weight more recent observations more heavily. An exponential weighting factor λ on the accumulated covariances can be included in the development of this algorithm. Typical values of λ are from .98 to 1.00 (corresponding to full weighting of past samples). The algorithms presented in subsequent sections include this exponential weighting factor.

5.4 Exact Least Squares Lattice Recursions

The recursions so far have focused on the predictor vectors, $\mathbf{A}_{i,T}$, $\mathbf{C}_{i,T}$ and $\mathbf{D}_{i,T}$. For a P-th order prediction filter, these recursions require $O(P^2)$ operations per time sample since all the predictor coefficients change in an order-update. However for the lattice structure, due to its orthogonalizing nature, only the $i-th$ reflection coefficient changes in the $i-th$ order-update. The exact least squares recursion can be written directly in terms of lattice filter variables which require only $O(P)$ operations to update per time sample.

The order-update recursions for the lattice filter variables, $f_{i,T}$ and $b_{i,T}$ was developed in (42-45). The order-updates for $\sigma^{f}_{i,T}$ and $\sigma^{b}_{i,T}$ are given in (39) and (41) and the time-updates are given in (53) and (55). The reflection coefficients defined above depend on $\Delta_{i+1,T+1}$ which is related to partial correlations. Here, the time-update for $\Delta_{i+1,T+1}$ () is required to augment the correlation for the new data sample. These updates also require the order-update of $\gamma_{i,T}$, determined in (47).

These recursions are seen to compute the sample cross-covariance of the forward and backward prediction errors, using the optimal weighting $1/(1 - \gamma_{i-1,T})$. From Section 4, the gradient lattice equations have the same form as above except that the exact recursive least squares solution calculates and uses the optimal weighting factor for the new data sample.

The complete set of order-update and time-update recursions to obtain the exact least squares lattice predictor (LSL) is presented in ALGORITHM 3. When starting the lattice filter, only the stages that receive data are executed until P data samples have been observed, ie. $min(T,P)$ filter stages are used where T is the data sample number.

ALGORITHM 3:

Recursive Least Squares Lattice (LSL)

Scalar case for exponentially weighted data

Input Parameters

P = maximum order of lattice
λ = exponential weighting factor (usually .98 to 1.0)
x_T = data sample at time T

Variables

$\Delta_{i,T}$ = partial autocorrelation coefficients
$\gamma_{i,T}$ = likelihood variable
$f_{i,T}$, $b_{i,T}$ = forward (backward) prediction errors

$\sigma^f_{i,T}$, $\sigma^b_{i,T}$ = forward (backward) prediction error covariances

$k^f_{i,T}$, $k^b_{i,T}$ = forward (backward) reflection coefficient

Initialization

$f_{0,0} = b_{0,0} = x_0$, $\sigma^f_{0,0} = \sigma^b_{0,0} = x_0^2$
$\Delta_{i,i} = 0$, $\gamma_{-1,i} = 0$, $1 \leq i \leq P$

ITERATION FOR EVERY NEW DATA SAMPLE

For data sample x_T and previous results Δ, γ, b , σ^f , σ^b

$$f_{0,T} = b_{0,T} = x_T$$
$$\sigma^f_{0,T} = \sigma^b_{0,T} = \lambda\ \sigma^f_{0,T-1} + x_T^2$$

For each stage of the lattice, $i = 0$, $min(T,P)$–1

$$\Delta_{i+1,T} = \lambda\ \Delta_{i+1,T-1} + b_{i,T-1}\, f_{i,T} / (1 - \gamma_{i-1,T-1})$$

$$\gamma_{i,T} = \gamma_{i-1,T} + b^2_{i,T} / \sigma^b_{i,T}$$

$$k^f_{i+1,T} = \Delta_{i+1,T} / \sigma^f_{i,T}$$

$$k^b_{i+1,T} = \Delta_{i+1,T} / \sigma^b_{i,T-1}$$

$$f_{i+1,T} = f_{i,T} - k^b_{i+1,T}\, b_{i,T-1}$$

$$b_{i+1,T} = b_{i,T-1} - k^f_{i+1,T}\, f_{i,T}$$

when $T \leq P$ $\quad \sigma^f_{i+1,T} = \sigma^f_{i,T} - k^b_{i+1,T}\, \Delta_{i+1,T}$

$$\sigma^b_{i+1,T} = \sigma^b_{i,T-1} - k^f_{i+1,T}\, \Delta_{i+1,T}$$

else $\quad \sigma^f_{i+1,T} = \lambda\ \sigma^f_{i+1,T-1} + f^2_{i+1,T} / (1 - \gamma_{i,T-1})$

$$\sigma^b_{i+1,T} = \lambda\ \sigma^b_{i+1,T-1} + b^2_{i+1,T} / (1 - \gamma_{i,T})$$

5.5 A Likelihood Variable

Information about changes in the nature of the observed process can be determined from $\gamma_{P,T}$. This variable can be interpreted as a sample data approximation of a statistical likelihood variable. For a zero mean Gaussian random process, z, the joint distribution for $\{z_T, z_{T-1}, \ldots, z_{T-P}\}$ is given by (58) with the covariance matrix as defined in (25).

$$p(z_T, \ldots, z_{T-P}) = |2\pi \mathbf{R_P}|^{-1/2} \exp\{-\frac{1}{2} \mathbf{x}^T_{|T:T-P|} \mathbf{R_P}^{-1} \mathbf{x}_{|T:T-P|}\} \quad (58)$$

The determinant of the covariance matrix, $|\mathbf{R_P}|$ is related to the reflection coefficients and the process variance R_0 [Markel and Gray, 1976].

$$|\mathbf{R_P}| = R_0 \prod_{i=1}^{P} (1 - K_i^2) \quad (59)$$

The logarithm of (58) becomes a log-likelihood function composed of two parts. The first two terms depend on the covariance of the process and the third term relies on the observed data samples.

$$\textit{log-likelihood} = \{\ln R_0 + \sum_{i=1}^{P} \ln(1 - K_i^2)\} + \mathbf{x}^T_{|T:T-P|} \mathbf{R_P}^{-1} \mathbf{x}_{|T:T-P|} \quad (60)$$

The variable $\gamma_{P,T}$ obtained in the exact least-squares recursions can be interpreted as the sample estimate of the third term in (60). The definition of $\gamma_{P,T}$ uses the sample estimate of the covariance matrix, $R_{P,T}$ instead of the known covariance matrix, R_P. Thus $\gamma_{P,T}$ is a measure of the likelihood that the P most recent data samples, $\{z_T, \ldots, z_{T-P}\}$ come from a Gaussian process with sample covariance $R_{P,T}$ determined from all of the past observations $\{z_j, 0 \le j \le T\}$. Since $0 \le \gamma_{P,T} \le 1$, a small value of $\gamma_{P,T}$ indicates that the recent data samples are likely observations from a Gaussian process with covariance $R_{P,T}$. However, a value of $\gamma_{P,T}$ near one implied that given the current Gaussian process assumption, the observations are unexpected; either the new observations come from a different Gaussian process due to a time varying nature of the physical process or there is a non-Gaussian component in the observations. So $\gamma_{P,T}$ can be used as a detection statistic for changes in the process characteristics or for unexpected (non-Gaussian) components in the observations. Simulations indeed demonstrated that $\gamma_{P,T}$ does take values

close to one when sudden changes in the observations occurred. In the LSL algorithm, $\gamma_{P,T}$ acts as an optimal gain control since the new observation influences the accumulated estimate by a factor of $(1 - \gamma_{P,T})^{-1}$. With this gain factor, changes in the process statistics can instantaneously influence the estimates more than just being averaged with past observations. Simulation results that demonstrate this behavior on synthetic signals and speech signals are shown in Section 9.

6. JOINT PROCESS LATTICE FILTER

Many practical problems require the joint interaction of two or more processes rather than the prediction of a process based on its own past observations. For example, a channel equalizer in its adaptation phase uses the received distorted signal from the channel and the actual transmitted channel symbols to determine the channel distortion characteristics. In a noise canceller, the information signal plus noise and a reference or noise signal is used to extract the information. The estimation of autoregressive moving average (ARMA) processes with known input involves the joint interaction of two processes. In general, a multi-channel problem can be formulated as a single vector process problem which can be solved by extending the previous least squares solution to the vector case. The one channel prediction lattice of Section 5 is extended here to include a second related channel. This joint process recursive least squares provides very fast tracking or adaptation for channel equalization or noise cancelling.

When one process y is to be estimated from observations of a related process x, it is possible to combine them into a joint process (x,y), that can be solved as a joint process lattice filter. The exact least squares solution for joint process estimation is an extension of the development in Section 5. A new prediction error is defined that includes samples from both processes. The joint prediction error, $j_{P,T}$ is the error in estimating y_T from $\{x_T,\ x_{T-1},\ \ldots,\ x_{T-P}\}$ where $\{g_i^P\}$ are the prediction coefficients obtained by minimizing the sum of the squared errors.

$$j_{P,T} = y_T + \sum_{i=0}^{P} g_i^P\, x_{T-i} \tag{61}$$

The solution of (61) can be formulated in terms of the lattice structure just as the single process predictor (32) was translated into (42). A prediction lattice filter (LSL) for x_T performs a Gram-Schmidt orthogonalization of $\{x_{T-i}\}$ into the mutually orthogonal backward prediction errors $\{b_{T-i}\}$. The advantage of using the orthogonal $\{b_{T-i}\}$ instead of $\{x_{T-i}\}$ in (61) is that the joint predictor coefficients $\{g_i^P\}$ become decoupled from one another so faster convergence is possible.

The joint process lattice solution involves the LSL for the x process and a similar lattice recursion of the joint prediction error, $j_{i,T}$. From the LSL for the x process, at the i-th lattice

stage, $b_{i,T}$ is the backward prediction error, $\sigma^{b}_{i,T}$ is its variance and the likelihood variable is $\gamma_{i-1,T}$. A new cross correlation term, $\Delta^{j}_{i,T}$, similar to (57), can be defined between signals available after the $i-th$ lattice stage, $j_{i-1,T}$ and $b_{i,T}$. This new term can be recursively updated.

$$\Delta^{j}_{i,T} = \lambda \Delta^{j}_{i,T-1} + \left[\frac{j_{i-1,T}\, b_{i,T}}{1 - \gamma_{i-1,T}}\right] \tag{62}$$

The recursion for the joint prediction error $j_{i,T}$ is similar to (42) except that quantities after the $i-th$ lattice stage are used. The initial condition is $j_{-1,T} = y_T$ and the output is $j_{i,T}$.

$$j_{i,T} = j_{i-1,T} - (\Delta^{j}_{i,T} / \sigma^{b}_{i,T})\ b_{i,T} \tag{63}$$

The previous single channel LSL equations augmented with (62) and (63) form the complete solution to the joint estimation problem and lead to the joint process lattice filter, Fig. 7.

For noise cancelling problems, noisy data containing the signal of interest, $\{y_T\}$ are observed together with the noise estimate or reference signal, $\{x_T\}$, see [Satorius et al, 1979(1), Griffiths, 1979(2)]. For channel equalization problems, $\{y_T\}$ is a known training sequence sent though the channel and $\{x_T\}$ is the distorted channel output. Applications of the joint process lattice estimation algorithms to adaptive data equalization have been investigated [Satorius and Alexander, 1979(2), Satorius and Pack, 1981].

The ARMA estimation problem with known input and with bootstrap estimated input was formulated as a two channel lattice filter in [Lee et al, 1982]. For an input process y, the output ARMA process is x generated in the following manner.

$$x_T + \sum_{i=1}^{P} a_i\, x_{T-i} = y_T + \sum_{i=1}^{P} b_i\, y_{T-i}$$

A prediction equation can be written for the process x, if the input y is considered known. This predictor follows from Section 5 but now includes a weighted combination of past inputs y.

$$\hat{x}_T = - \sum_{i=1}^{P} a_i\, x_{T-i} + \sum_{i=1}^{P} b_i\, y_{T-i}$$

Similarly, a prediction equation for the y process with x known is generated by extending (61) to include a weighted combination of past inputs y.

$$\hat{y}_T = - \sum_{i=1}^{P} c_i\, y_{T-i} + \sum_{i=1}^{P} d_i\, x_{T-i}$$

The vector process $[x_T, y_T]$ has a structure similar to the scalar AR processes discussed earlier. Prediction errors are now vectors and covariances are matrices. The ARMA lattice estimation algorithm follows the scalar LSL or SQNLSL algorithm but the quantities are now vectors and matrices. The reflection coefficient has become a two by two matrix. Further details are found in [Lee et al, 1982].

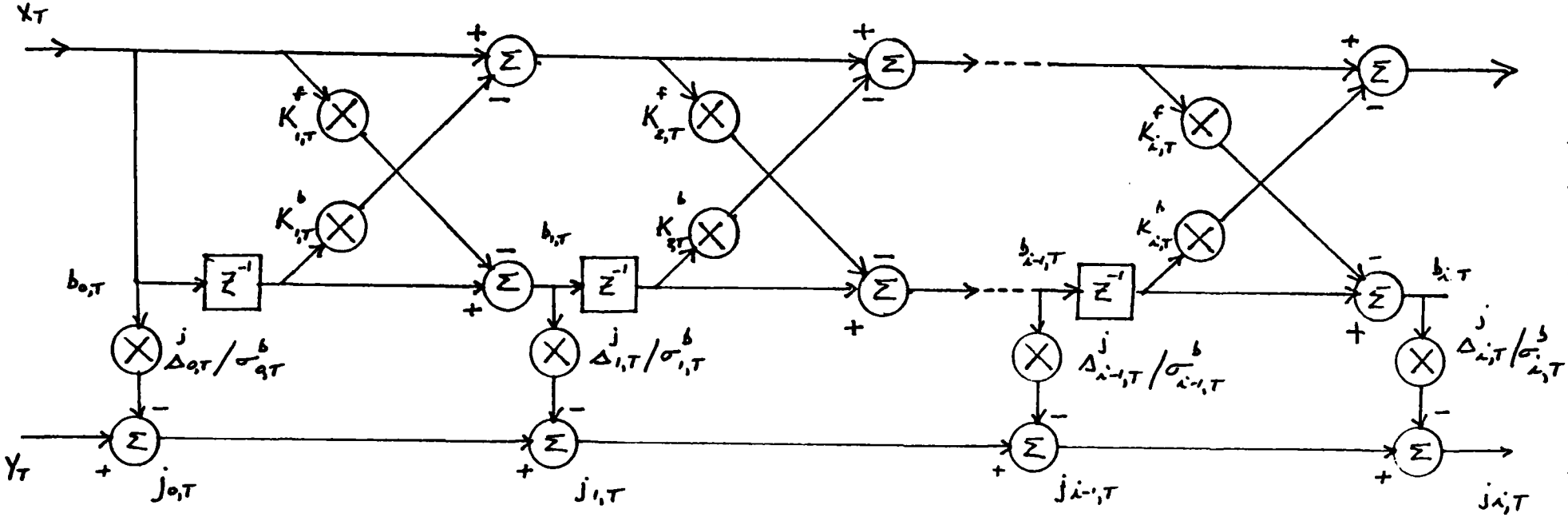


Fig. 7 Joint Process Lattice Filter

7. SQUARE ROOT NORMALIZED LEAST SQUARES LATTICE

The complexity of the recursions can be reduced and the numerical properties of the variables improved by rewriting the Least Squares Lattice algorithm in terms of normalized variables. The Square Root Normalized Least Squares Lattice, (SQNLSL) developed in [Lee, 1980] has only three recursions per order for each time sample where all three variables have unit variance. The reduction of the LSL into the SQNLSL requires two types of normalizations. A variance normalization scales the variables by their respective variances. A normalization by the optimal weighting factor using the likelihood variable, γ is also necessary. A brief development of the SQNLSL is presented here; see [Lee et al, 1981] for more details.

The forward and backward prediction errors when normalized become $\nu_{i,T}$ and $\eta_{i,T}$ respectfully. The normalizing factors are the square roots of the variances, $\sigma^f_{i,T}$ and $\sigma^b_{i,T}$ and the square root of the optimal weighting factor $(1 - \gamma_{i-1,T-1})$.

$$\nu_{i,T} = f_{i,T} (\sigma^f_{i,T})^{-1/2} (1 - \gamma_{i-1,T-1})^{-1/2} \tag{64}$$
$$\eta_{i,T-1} = b_{i,T-1} (\sigma^b_{i,T-1})^{-1/2} (1 - \gamma_{i-1,T-1})^{-1/2}$$

By combining the two reflection coefficients from the LSL, the normalized partial correlation, $\rho_{i,T}$ is defined like a correlation coefficient. This single parameter is the new reflection coefficient.

$$\rho_{i+1,T} = (\sigma^f_{i,T})^{-1/2} \Delta_{i+1,T} (\sigma^b_{i,T-1})^{-1/2} \tag{65}$$

First the recursion for the partial correlation (65) will be developed from the LSL algorithm. The variances of the prediction errors (with exponential weighting λ) has a time-update recursions given by (53).

$$\lambda \sigma^f_{i,T-1} = \sigma^f_{i,T} - \frac{f^2_{i,T}}{1 - \gamma_{i-1,T-1}}$$

By dividing by $\sigma^f_{i,T}$ and using the definition for $\nu_{i,T}$, the time-update for the variance can be related to the new variables (66). A similar relation (67) exists for the backward prediction error variance.

$$\lambda \sigma^f_{i,T-1} / \sigma^f_{i,T} = 1 - \nu^2_{i,T} \tag{66}$$
$$\lambda \sigma^b_{i,T-1} / \sigma^b_{i,T} = 1 - \eta^2_{i,T} \tag{67}$$

The time-update recursion for the normalized partial correlations are obtained by substituting the

expression for $\Delta_{i+1,T}$ (57) (including the exponential weighting λ) in (65).

$$\Delta_{i+1,T} = \Delta_{i+1,T-1} + \frac{b_{i,T-1}\, f_{i,T}}{1 - \gamma_{i-1,T}}$$

$$\rho_{i+1,T} = (\sigma^f_{i,T})^{-1/2} \left(\lambda \Delta_{i+1,T-1} + \frac{b_{i,T-1}\, f_{i,T}}{1 - \gamma_{i-1,T-1}} \right) (\sigma^b_{i,T-1})^{-1/2} \tag{68}$$

The first term is replaced by the definition of $\rho_{i+1,T-1}$ and the second term is $\nu_{i,T}\, \eta_{i,T-1}$.

$$\rho_{i+1,T} = \lambda (\sigma^f_{i,T})^{-1/2} (\sigma^f_{i,T-1})^{1/2} \rho_{i+1,T-1} (\sigma^b_{i,T-2})^{-1/2} (\sigma^b_{i,T-1})^{1/2} + \nu_{i,T}\, \eta_{i,T-1}$$

Using (66) and (67), the new time-update equation for the normalized partial correlation simplifies to (69).

$$\rho_{i+1,T} = (1 - \nu^2_{i,T})^{1/2} \rho_{i+1,T-1} (1 - \eta^2_{i,T-1})^{1/2} + \nu_{i,T}\, \eta_{i,T-1} \tag{69}$$

Now the lattice recursions can be written in terms of these new variables. The order-update recursions for the forward prediction errors (42) can be written using the normalized partial correlation.

$$(1 - \gamma_{i,T-1})^{1/2} \nu_{i+1,T} (\sigma^f_{i+1,T})^{1/2} = (\sigma^f_{i,T})^{1/2} (\nu_{i,T} - \rho_{i+1,T}\, \eta_{i,T-1}) (1 - \gamma_{i-1,T-1})^{1/2} \tag{70}$$

To simplify this expression, two order-update equations from the development of LSL are needed; for the likelihood variable (47) and for the prediction error variances.

$$(1 - \gamma_{i,T}) = (1 - \gamma_{i-1,T}) (1 - \eta^2_{i,T})$$

$$\sigma^f_{i+1,T} / \sigma^f_{i,T} = 1 - \rho^2_{i+1,T}$$

Using these relations, (70) can be reduced to a simple expression for the normalized forward prediction errors (71). A similar development for the backwards prediction error leads to (72).

$$\nu_{i+1,T} = (1 - \rho^2_{i+1,T})^{-1/2} (\nu_{i,T} - \rho_{i+1,T}\, \eta_{i,T-1}) (1 - \eta^2_{i,T-1})^{-1/2} \tag{71}$$

$$\eta_{i+1,T} = (1 - \rho^2_{i+1,T})^{-1/2} (\eta_{i,T-1} - \rho_{i+1,T}\, \nu_{i,T}) (1 - \nu^2_{i,T})^{-1/2} \tag{72}$$

The lattice recursions have now become three equations, (69), (71), and (72) that compute the normalized prediction errors, $\{\nu\}$ and $\{\eta\}$, and the reflection coefficients, $\{\rho\}$ for each lattice stage and for every data sample. Proper initialization is required to start the recursions with unit variance quantities.

The reflection coefficients in the SQNLSL still have magnitudes bounded by one but now the prediction errors are also bounded. The complexity of the lattice recursions has been reduced from six recursions to only three recursions per order and time update. Square root operations are required. They can be efficiently computed by bit recursive algorithms such as the CORDIC technique (discussed in the next section). Simple recursions and their potentially better numerical behavior, makes the SQNLSL algorithm preferable over the unnormalized LSL version.

The SQNLSL just developed applies for exponentially weighted data. However the exponential weight λ in (68) is not evident in these recursions. By combining the three time-update recursions for $\sigma_{i,T}^{f}$, $\sigma_{i,T}^{b}$ and $\Delta_{i+1,T}$ into one recursion for $\rho_{i+1,T}$, the effect of λ is carried through unseen. When a new data sample is used, the exponential weighting is applied to the sample variance estimate. ALGORITHM 4 summarizes the Square Root Normalized Least Squares Lattice (SQNLSL) estimation method. The sample variance R_T is initialized to some value σ to avoid dividing by zero.

Although SQNLSL is a very powerful and compact algorithm, the necessity of computing square roots can lead to problems. The fixed point error analysis of this algorithm [Samson and Reddy] indicated that finite wordlength arithmetic computation of the square roots lead to small biases in the reflection coefficients. This bias was more predominant than the variance of the error in the estimate and generally quite small. The bias increased as the wordlength became shorter or the exponential weighting factor λ approached one.

ALGORITHM 4:

Square Root Normalized Least Squares Lattice (SQNLSL)

Exponentially Weighted Scalar Data

Input Parameters

P = maximum order of lattice filter
λ = exponential weighting factor (usually .98 to 1.0)
σ = prior variance
x_T = data sample at time T

Variables

R_T = estimated variance of x
$\rho_{i,T}$ = reflection coefficients
$\nu_{i,T}$ = normalized forward prediction error
$\eta_{i,T}$ = normalized backward prediction error

Initialization

$$R_o = \sigma + x_o^2$$
$$\nu_{o,o} = \eta_{o,o} = x_o/\sqrt{R_o}$$
$$\rho_{i,o} = 0 \quad 1 \leq i \leq P$$

ITERATION FOR EVERY NEW DATA SAMPLE

New data sample x_T and previous results $\eta_{i,T-1}$, $\rho_{i+1,T-1}$, R_{T-1}

$$R_T = \lambda R_{T-1} + x_T^2$$

$$\nu_{o,T} = \eta_{o,T} = x_T/\sqrt{R_T}$$

For each stage of the lattice, $i = 0$ to $min(T,P)-1$

$$\rho_{i+1,T} = \sqrt{1-\nu_{i,T}^2}\sqrt{1-\eta_{i,T-1}^2}\,\rho_{i+1,T-1} + \nu_{i,T}\eta_{i,T-1}$$

$$\nu_{i+1,T} = \frac{\nu_{i,T} - \rho_{i+1,T}\eta_{i,T-1}}{\sqrt{1-\rho_{i+1,T}^2}\sqrt{1-\eta_{i,T-1}^2}}$$

$$\eta_{i+1,T} = \frac{\eta_{i,T-1} - \rho_{i+1,T}\nu_{i,T}}{\sqrt{1-\rho_{i+1,T}^2}\sqrt{1-\nu_{i,T}^2}}$$

8. COMPUTATIONAL COMPLEXITY AND CORDIC ARITHMETIC

The complexity of the lattice filter is greater than the equivalent tapped delay line filter, However the lattice filter has many advantageous properties not shared by the simpler filter. Similarly the adaptive lattice filter has a few more operations than an adaptive tapped delay line filter. Table 1 compares the computational complexity of several adaptive estimation algorithms. The lattice methods require three to six times as many computations as the simplest adaptive tapped delay line filter (LMS). However this increase in computational complexity provides for substantially faster convergence, better numerical properties of the coefficients, and an assurance of a stable filter. The complexity of several adaptive algorithms is presented in the following table. The scaling by a constant weighting factor, eg. λ or β is usually approximated as a shifting by a power of two. Thus this fixed scaling is not included in the count of operations. The LMS algorithm is the tapped delay line gradient least mean squares technique. The gradient lattice algorithms is (22) and (18). ALGORITHM 3 is denoted LSL and ALGORITHM 4 is called SQNLSL. The number of operations are counted for executing a single filter stage on a single data sample. To process T data samples in an N-th order filter would require NT time as many computations.

TABLE 1 Computational Complexity

Algorithm	×	÷	√	±
LMS	2	0	0	2
Grad. Latt.	6	1	0	6
LSL	6	6	0	7
SQNLSL	10	2	3	6

The SQNLSL algorithm is the most complex of the algorithms, requiring three square roots, ten multiplications, and two divisions to execute an update for each stage in the lattice for every new data sample. However, the SQNLSL has a very compact form with only three equations

involving variables that have constrained magnitudes. The implementation of the SQNLSL equations in hardware would require special multiplier hardware for fast execution or would require considerable execution time on general purpose microprocessors. Even using shift and add instead of multiplication and Newton's method for square root operations would require considerable execution time.

By interpreting the lattice equations as rotations, an efficient realization of the square root normalized lattice algorithm using CORDIC arithmetic was developed in [Ahmed, 1981(1)]. The COordinate Rotation DIgital Computer (CORDIC) developed in [Volder] is an iterative algorithm for computing trigonometric functions, multiplications, divisions, and square roots. The CORDIC algorithm interprets the above functions as rotations of a vector in different coordinate systems. The rotation is implemented by a sequence of shift and add operations. This type of arithmetic is not new; it has been used to compute trigonometric functions and their inverses in hand held calculators. Many other signal processing algorithms, such as DFT and matrix inversion, can also be implemented in arrays of CORDIC processors [Ahmed et al, 1982].

8.1 CORDIC Arithmetic

The well known equation for rotating a vector $[x_i, y_i]^T$ to a new vector $[x_{i+1}, y_{i+1}]^T$ uses the *sine* and *cosine* of the rotation angle θ.

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

Four multiplications by two trigonometric quantities are required. This operation can be made more amenable to fast computer implementation by modifying this equation into a sequence of small rotations of a specific form, each implemented using only additions and shifts.

CORDIC arithmetic was unified into a single equation [Walther] that allows rotations on either a circle ($m=1$), along a line ($m=0$), or along a hyperbola ($m=-1$). The incremental unit of rotation at the i-th iteration is the predetermined sequence $\{\delta_i\}$ and $\mu_i = \pm 1$ that determines the direction of rotation.

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & m\mu_i\delta_i \\ -\mu_i\delta_i & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{78}$$

The $\{\delta_i\}$ must be chosen to satisfy certain constraints that assure convergence of the iterations. To obtain computational efficiency on current computer hardware using binary representations, a negative integer power of two is chosen for δ_i. This allows the multiplication by δ_i to be performed as a shift.

The effect of (78) is a rotation and a change of scale interpreted in the appropriate coordinate space. The vector $[x_0, y_0]^T$ can be represented as a generalized radial component R_0 and a generalized angular component, Φ_0.

$$R_0 = \sqrt{x_0^2 + my_0^2}$$
$$\Phi_0 = \sqrt{m}\ \tan^{-1}(y_0\sqrt{m}\,/x_0)$$

For the circular rotation, this is a true polar coordinate representation. Performing the operation in (78) scales the radial component by $r_i = \sqrt{1 + m\delta_i^2}$ and changes the angular component by $\phi_i = m^{-1/2}\ \tan^{-1}(\delta_i\sqrt{m})$. After p iterations, the new radial and angular components are R_p and Φ_p.

$$R_p = R_0 \prod_{i=1}^{p} r_i = R_0 \prod_{i=1}^{p} \sqrt{1 + m\delta_i^2} \tag{79}$$
$$\Phi_p = \Phi_0 - \sum_{i=1}^{p} \mu_i \phi_i = \Phi_0 - \sum_{i=1}^{p} \mu_i\ m^{-1/2} \tan^{-1}(\delta_i\sqrt{m})$$

The convergence of the iterations and the efficient implementation of the rotations depend critically on the predetermined choice of δ_i. Each type of rotation, $(m = -1,0,+1)$ has a different predetermined set of positive increments (δ_i) which specify fixed radial and angular increments (r_i, ϕ_i) from (79). Within the domain of convergence (limited by the total possible rotation) constraints were developed on the sequence $\{\phi_i\}$ such that any angle Φ could be rotated to within ϕ_{p-1} of zero in p steps [Walther]. This guarantees that the granularity of the calculation (the angular resolution) is ϕ_{p-1} in p steps. With the proper choice of the set of increments $\{\delta_i\}$, each successive iteration yields approximately one more bit of accuracy in the final result.

The CORDIC equation, (78) is augmented by an additional variable, z_p that accumulates the angular component of the rotation.

$$z_p = z_0 - \sum_{i=1}^{p} \mu_i \phi_i \tag{80}$$

The three parameters (x,y,z) are manipulated by successive application of the CORDIC equation. The function to be computed is obtained by forcing either y or z to zero. Often the initial value of the other variable, z or y is zero or one. The sign of the rotation, μ_i is chosen at each iteration to move the desired parameter towards zero. Rotation operations are obtained by forcing z_p to zero. and vectoring operations result if y_p is forced to zero.

The interpretation of (78) as a rotation on a circle, line and hyperbola can be seen from a graphical perspective. For example, a circular rotation (m=1) of a vector $[x_o, y_o]^T$ into the vector $[x_p, 0]^T$ computes $\tan^{-1} y_o/x_o$. The direction of rotation μ_i is chosen at each iteration to force y_i closer to zero. The sequence of small angular step, $\{\phi_i\}$, predetermined by $\{\delta_i\}$, is accumulated with appropriate sign in z_p, giving the answer. Fig. 8a indicates how this rotation proceeds. The radius of the circle increases a predetermined amount r_i with each rotation step. It is not necessary to account for this change in radius when computing $\tan^{-1} y_o/x_o$. However the value of x_p has become a scaled square root where the scale factor is known in advance.

$$x_p = \sqrt{x_0^2 + y_0^2} \cdot \prod_{i=1}^{p} r_i$$

The three input/output box of Fig. 8a is used to describe the function evaluated by the CORDIC rotation. With a nonzero initial value of z_0, forcing z_i to zero generates $\sin(z_0)$ and $\cos(z_0)$, see Fig 8a.

For a rotation on a line, the radial component is always one and the angle component is interpreted as the y_i value. The increments become $r_i = 1$ and $\phi_i = \delta_i$. The result of applying the CORDIC operation is shown in Fig. 8b. Multiplication and division can be calculated this way.

The hyperbolic rotation computes *sinh*, *cosh*, *arctanh* and square roots, see Fig. 8c. The surface of rotation is the set of point that is a constant distance $\sqrt{x^2-y^2}$ from the origin. This hyperbola moves a fixed amount, $\sqrt{1-\delta_i^2}$ closer to the origin after each CORDIC operation.

8.2 Lattice Filtering by Rotations

The square root normalized lattice equations have a natural interpretation as rotations. The three recursive equations can be efficiently realized using CORDIC arithmetic [Ahmed et al, 1981(1), 1981(2)]. The structure of the SQNLSL algorithm suggests an implementation via rotations. Since ν, ρ, and η always have magnitudes less than one, they can be interpreted as cosines of angles. Furthermore, if $x = cos(\theta_x)$, then the complement of x is $x^c = \sqrt{1 - x^2} = \sin(\theta_x)$. The SQNLSL equations can be written using a compact notation.

$$\rho_{i+1,T} = \sqrt{1-\nu_{i,T}^2}\sqrt{1-\eta_{i,T-1}^2}\,\rho_{i+1,T-1} + \nu_{i,T}\eta_{i,T-1} \qquad \rightarrow \qquad \rho_+ = \nu^c\eta^c\rho + \nu\eta$$

$$\nu_{i+1,T} = \frac{\nu_{i,T} - \rho_{i+1,T}\eta_{i,T-1}}{\sqrt{1-\rho_{i+1,T}^2}\sqrt{1-\eta_{i,T-1}^2}} \qquad \rightarrow \qquad \nu_+ = (\nu - \rho_+\eta) \,/\, \rho_+^c\eta^c$$

$$\eta_{i+1,T} = \frac{\eta_{i,T-1} - \rho_{i+1,T}\nu_{i,T}}{\sqrt{1-\rho_{i+1,T}^2}\sqrt{1-\nu_{i,T}^2}} \qquad \rightarrow \qquad \eta_+ = (\eta - \rho_+\nu) \,/\, \rho_+^c\nu^c$$

For notational convenience, the following abbreviations were used.

$$\rho = \rho_{i+1,T-1}\,, \qquad \rho_+ = \rho_{i+1,T}$$
$$\nu = \nu_{i,T}\,, \qquad \nu_+ = \nu_{i+1,T}$$
$$\eta = \eta_{i,T-1}\,, \qquad \eta_+ = \eta_{i+1,T}$$

The SQNLSL update equations can be written almost entirely in a single matrix equation (81) using the rotation matrices for ν and η.

$$\begin{bmatrix} \nu^c & \nu \\ -\nu & \nu^c \end{bmatrix} \begin{bmatrix} \rho & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta^c & -\eta \\ \eta & \eta^c \end{bmatrix} = \begin{bmatrix} \nu^c\eta^c\rho + \nu\eta & \nu\eta^c - \rho\eta\nu^c \\ \nu^c\eta - \rho\nu\eta^c & \nu^c\eta^c + \rho\nu\eta \end{bmatrix} = \begin{bmatrix} \rho_+ & \rho_+^c\nu_+ \\ \rho_+^c\eta_+ & * \end{bmatrix} \qquad (81)$$

On the left hand side of (81), the first matrix performs a rotation by $\theta_\nu = \cos^{-1}(\nu)$ and the third matrix rotates by $\theta_\eta = \cos^{-1}(\eta)$. The result is the complete update for ρ and partial updates for ν and η and a term ($*$) of no interest. The updates for ν and η are completed by dividing by ρ_+^c. This matrix equation (81) is directly realizable using CORDIC operations.

The implementation of the SQNLSL algorithm in an integrated circuit proposes using two processors in parallel, each executing sequentially five functions, [Ahmed et al, 1981(1), 1981(2)].

The computation proceeds from (81) in three step; an angular rotation by θ_ν, the multiplication of ρ matrix by the η rotation matrix, and the divisions by ρ_+^c. The sequence of CORDIC operations shown in Fig. 10 begins with ρ, η and ν and computes ρ_+,η_+ and ν_+. The functional elements use the notation of Fig. 9. The rotation angle θ_ν is calculated as $\tan^{-1}(\nu/\nu^c)$ using a circular CORDIC operation by processor 2 in time slots 1 and 2. Rotating the ρ matrix by θ_η is computed as two multiplications (linear CORDIC): $\rho\,\eta^c$ by processor 1 during time slots 1 and 2, $\rho\,\eta$ by processor 2 in time slot 3. These quantities are rotated by θ_ν using circular CORDIC operations in processor 1 (time slot 3) and processor 2 (time slot 4). This generates the ρ update and partial updates for ν and η. In time slot 5, the processors generate the updates for ν and η by dividing the earlier results by ρ_+^c. The signals that flow farther than adjacent time slots must be held in temporary buffers. Each CORDIC operation uses 16 iterations and results in almost 16 bits of accuracy. The integrated circuit could perform the SQNLSL algorithm of tenth order on an 8 KHz. sampled signal in real time. This assumes standard integrated circuit design rules to generate a moderate size chip running at a 20 MHz. clock rate.

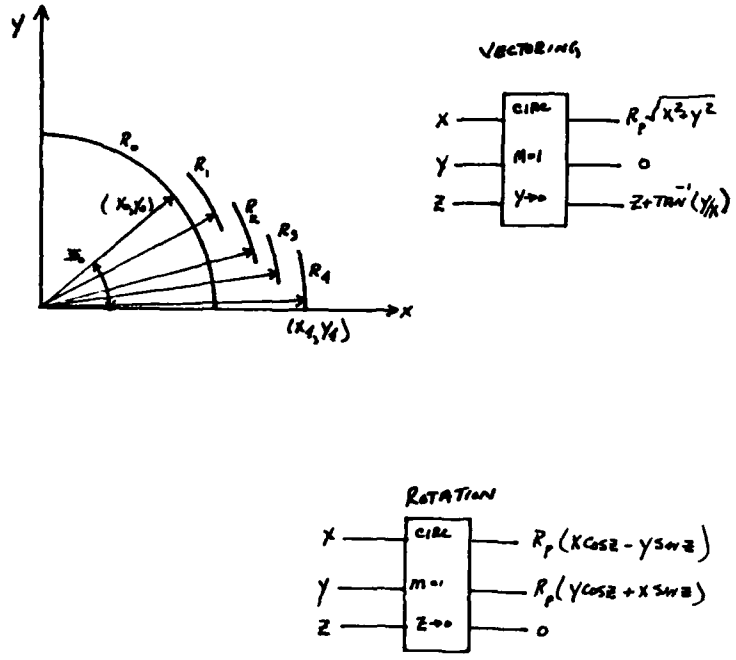


Fig. 8a CORDIC Rotation on a Circle

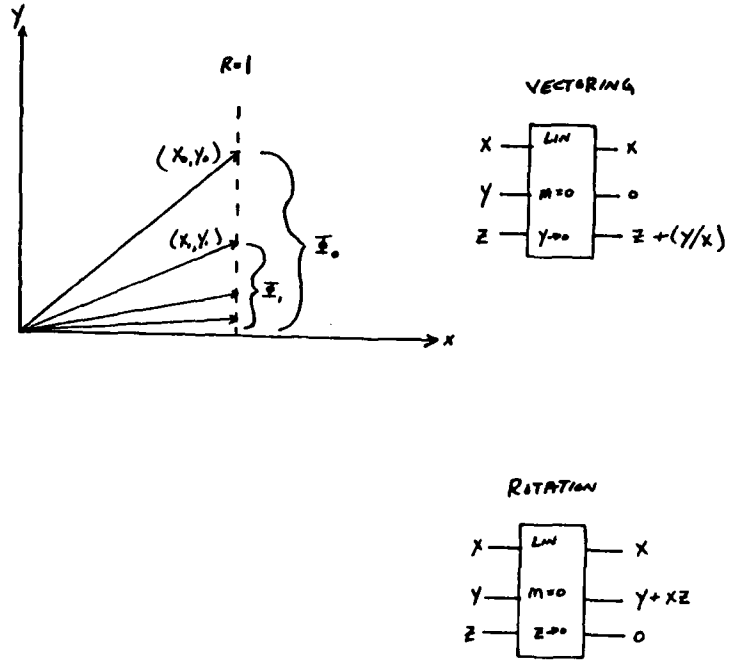
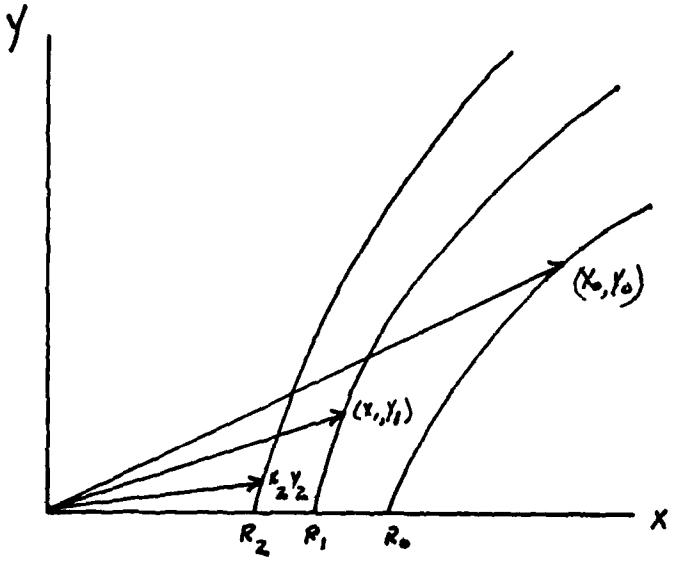
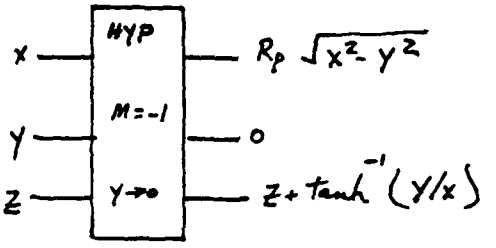


Fig. 8b CORDIC Rotation on a Line



VECTORING



ROTATION

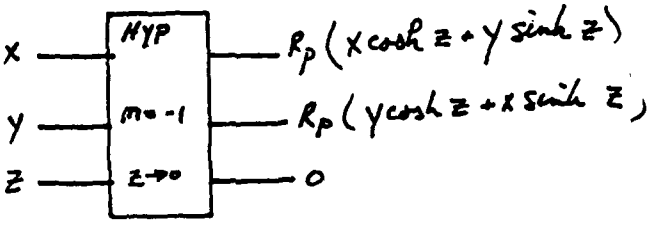


Fig. 8c CORDIC Rotation on a Hyperbola

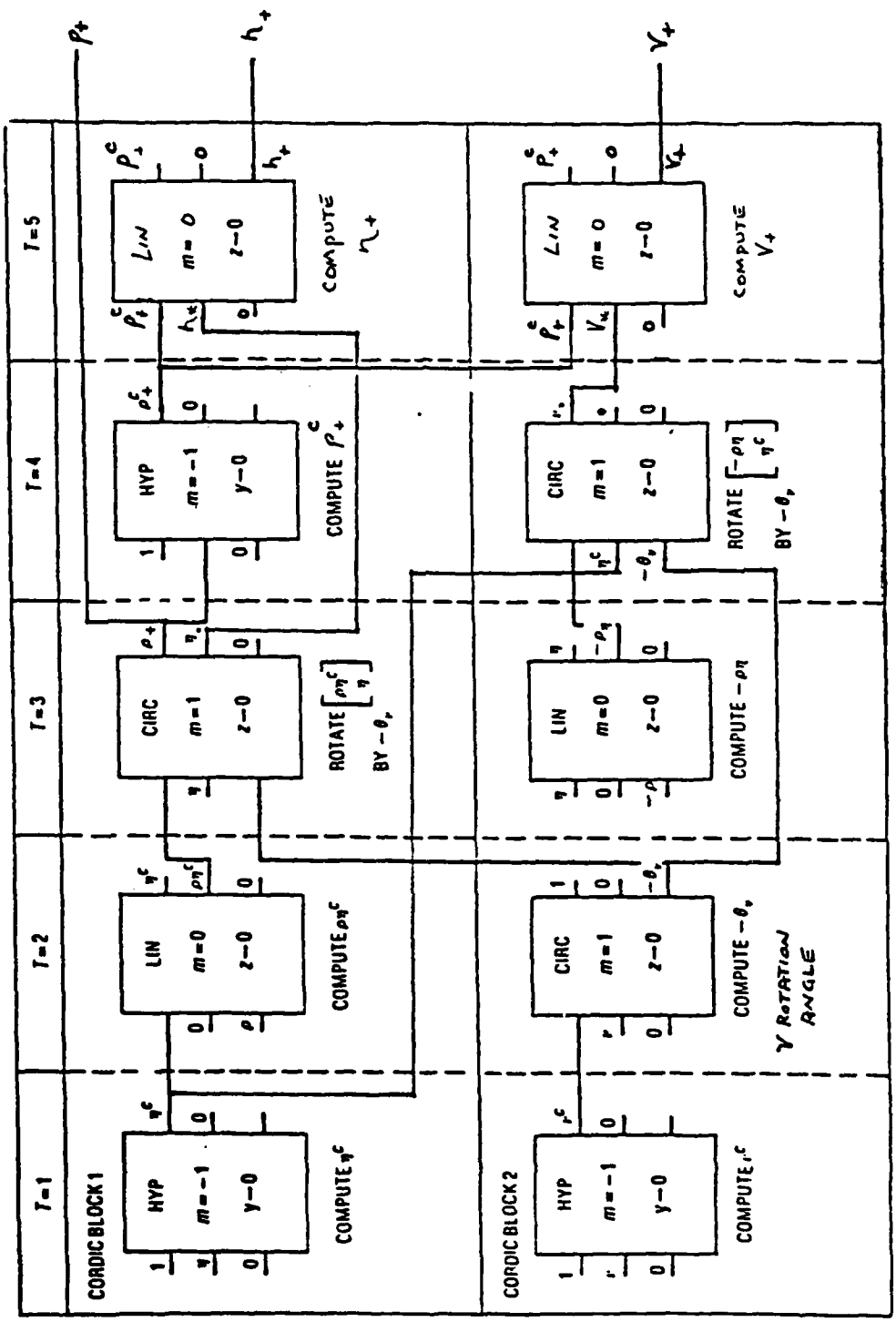


Fig. 9 CORDIC Implementation of Square Root Normalized Lattice

9. SIMULATIONS AND APPLICATIONS

SIMULATED SIGNALS

Simulated signals with different characteristics were generated and the response of the lattice estimation algorithms noted. The simulated signals included white noise drive autoregressive processes that were (1) stationary, (2) had linearly time varying coefficients, (3) had step changes in coefficients, and (4) had impulse excitation at the step changes in coefficients.

For a stationary autoregressive process, the convergence of the least squares lattice method is shown in Fig. 10. An eighth order fixed coefficient lattice filter driven by white noise generated the simulation data. The LSL algorithm with λ=.99 was used to compute the reflection coefficients. The first reflection coefficient converged in less than 50 samples and the first four reflection coefficients were near their correct values after 150 samples. Higher order reflection coefficients approached their correct values after 250 samples.

When the simulated data was generated by a white noise driven second order lattice with linearly time varying coefficients, the adaptive nature was be seen in Fig. 11. The two reflection coefficient estimates followed the actual parameter values. However, there was an increase in the variance of the estimate as the reflection coefficients approached zero or as the coefficient index increased. The previous experiment was repeated with piecewise constant coefficients to generate the simulation data. The estimated reflection coefficient trajectory did not indicate that the model had step changes in the coefficients, see Fig. 12.

The effect of the optimal weighting function γ was seen when the simulated data was generated by the same lattice with step changes in coefficients but also had a periodic impulse added to the white noise driving process at the instant of coefficient change. The presence of the impulses caused the estimates to readjust quickly to the new piecewise constant values, see Fig. 13. The impulse caused a sudden increase in the γ which allowed the estimates to focus on the new signal characteristics. Once the effect of the impulse has passed, the γ decreased so that convergence could take place.

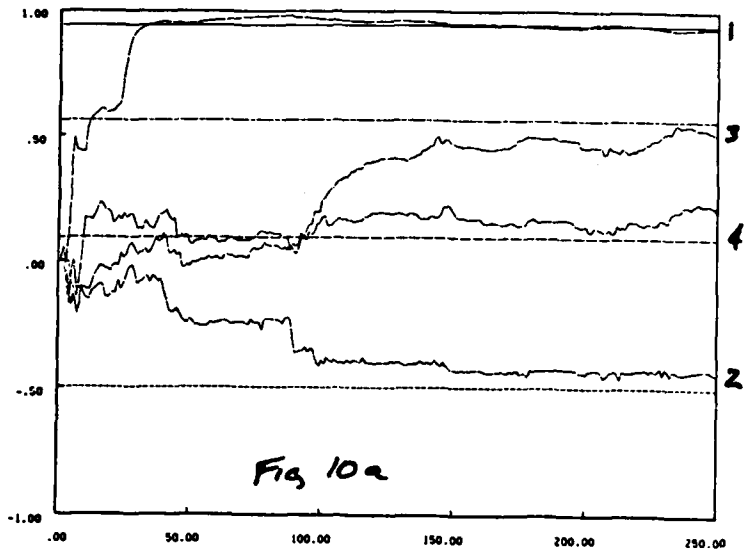


Fig 10a

CASE 1: Reflection coefficients of ULS, orders 1- 4

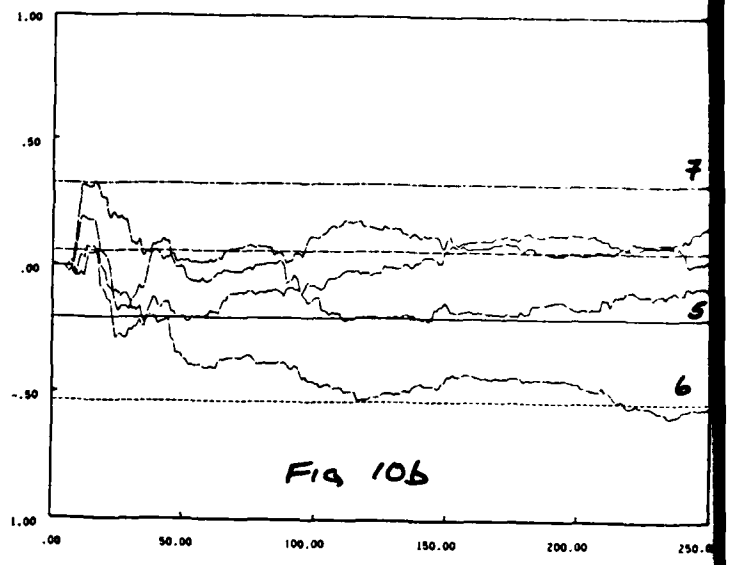


Fig 10b

CASE 1: Reflection coefficients of ULS: orders 5- 8

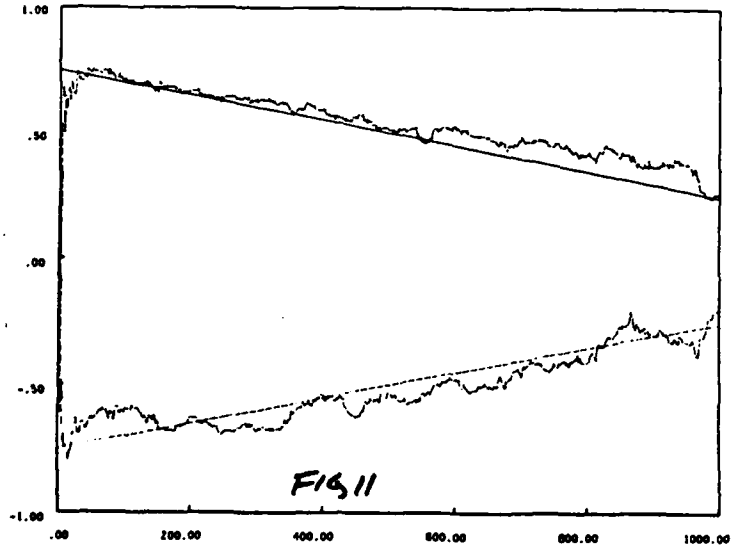


FIG 11

CASE 2: Reflection coefficients of ULS, orders 1- 2

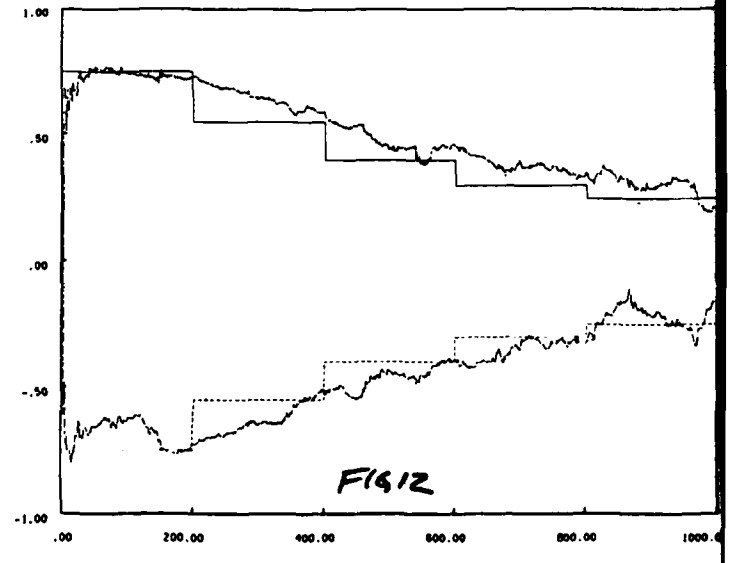


FIG 12

CASE 3: Reflection coefficients of ULS, orders 1- 2

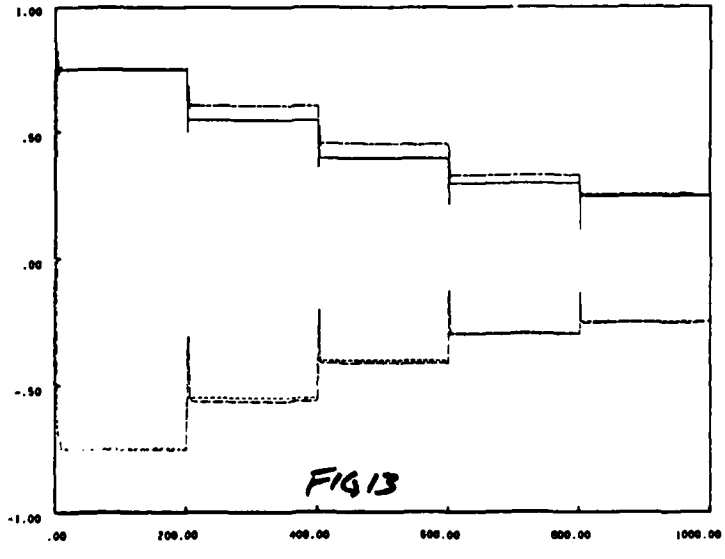


FIG 13

CASE 4: Reflection coefficients of ULS, orders 1- 1

APPLICATION FOR SPEECH ANALYSIS

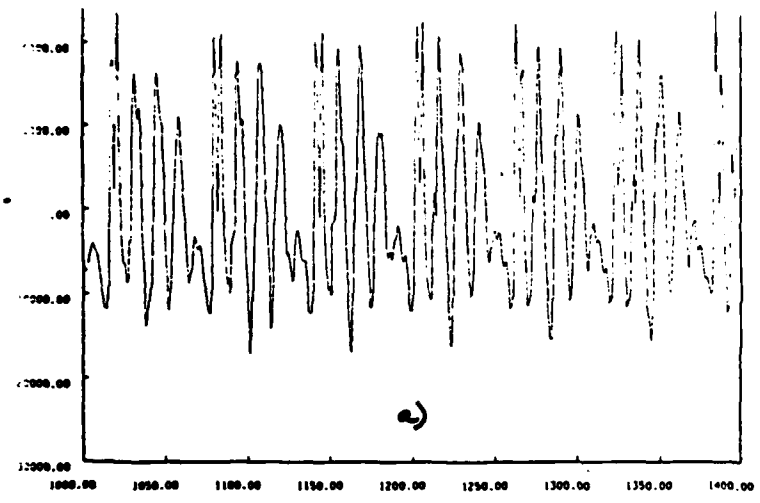
The most extensive use of the lattice filter has been for speech processing applications, including speech compression systems and stored vocabulary speech synthesis chips. Reflection coefficients and the lattice filter are well suited for speech processing for many reasons; the relationship to the acoustical tube model of the vocal tract, the advantageous quantization properties of the reflection coefficients, the finite word length arithmetic properties of the lattice filter, and the slowly time varying nature of the reflection coefficients across speech sounds (making them amenable to interpolation).

Linear Predictive Coding (LPC) is a technique that has been used widely for low bit rate speech coding and fixed vocabulary speech synthesis. LPC uses a vocal tract model, the lattice filter parameterized by reflection coefficients and an excitation model, periodic pulses for sounds produced by vocal chord oscillation (eg. vowels) and white noise for hiss sounds. Short time segments of speech, typically 20 milliseconds, are characterized by eight to ten reflection coefficients, the pulse period (zero for noise), and an energy term. All of the parameters can be quantized to a total of 48 bit per 20 millisecond interval. Using this compact description of sounds, speech synthesis integrated circuit have been developed that generate understandable speech using parameters store in read only memory.

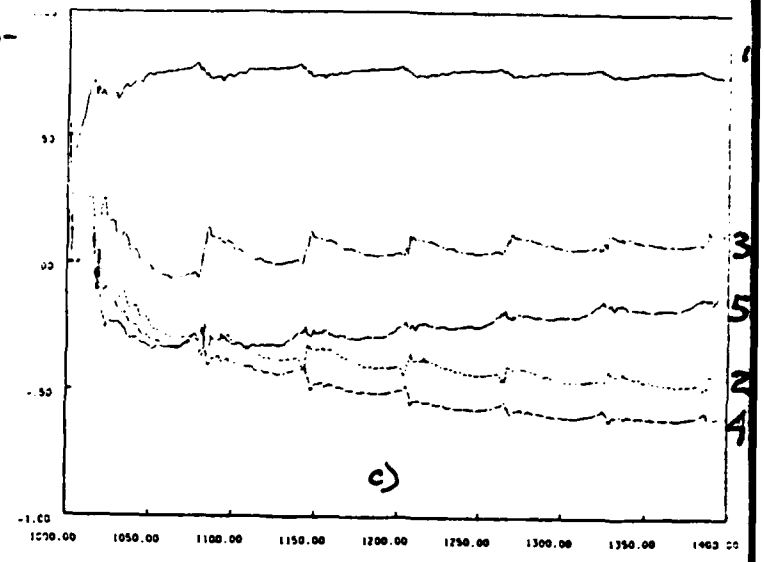
Analyzing a spoken vowel sound by the LSL algorithm shows the properties of the likelihood variable. The time waveform, Fig. 14a clearly shows the periodic nature of this vowel. The LSL algorithm applied to this sound produces the forward prediction error shown in Fig. 14b. This relatively stationary sound produced fairly constant reflection coefficients (after convergence), see Fig. 14c. The periodic jumps seen in all five reflection coefficients are due to the influence of the periodic opening of the vocal chords. The likelihood variable, γ usually is small but increases when these openings occur, Fig 14d. When the vocal chords open, a sudden pulse of air excites the vocal tract which the likelihood variable interprets as a change in the structure of the signal. Determining the periodicity of these openings, called the pitch period is necessary of the LPC speech model. Pitch pulses can be located directly from the prediction errors but do not

always give accurate results. The periodicity is evident but not easily extracted from Fig. 14b. By combines the derivative of the likelihood variable γ with the prediction error sequence, a more easily discernible spike is generated at the onset of vocal chord oscillation, see Fig. 14e. This technique has been proposed as a pitch estimation method [Lee and Morf, 1980].

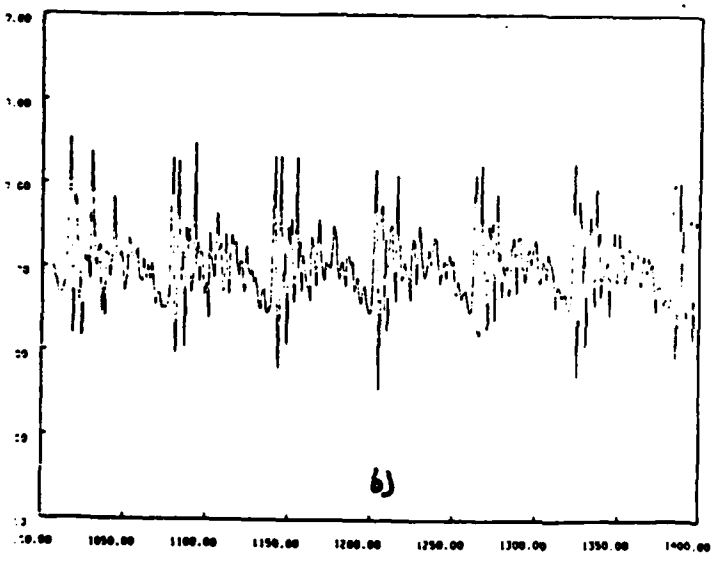
Since the recursive exact least squares lattice algorithms can track quickly changing spectral characteristics, they can be used to differentiate the nature of transitional sounds [Turner, 1982]. By exponential weighting of past data, the current estimate reflects the short time signal characteristics. The beginnings of the words 'bid' and 'did' spoken by the same male speaker are shown in Fig. 15. When analyzed by the SQNLSL with $\lambda = .98$, the reflection coefficients for the beginning of each word follow different trajectories corresponding to the different consonants. However, during the later vowel portion, the values are more similar, see Fig. 15b,c,e,f. The transitional part of the sounds is emphasized but the effects of the pitch pulses are also seen. The ability of the SQNLSL to differentiate these types of sounds may be useful in a phoneme based speech recognition systems.



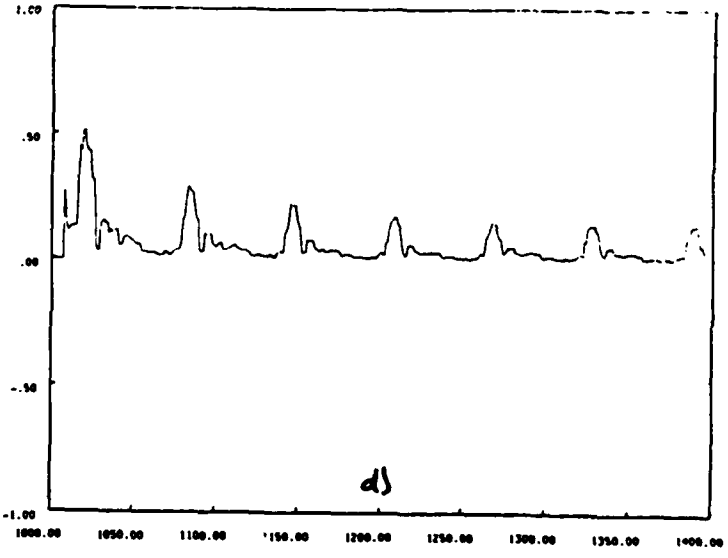
16-bit digitized speech- file8



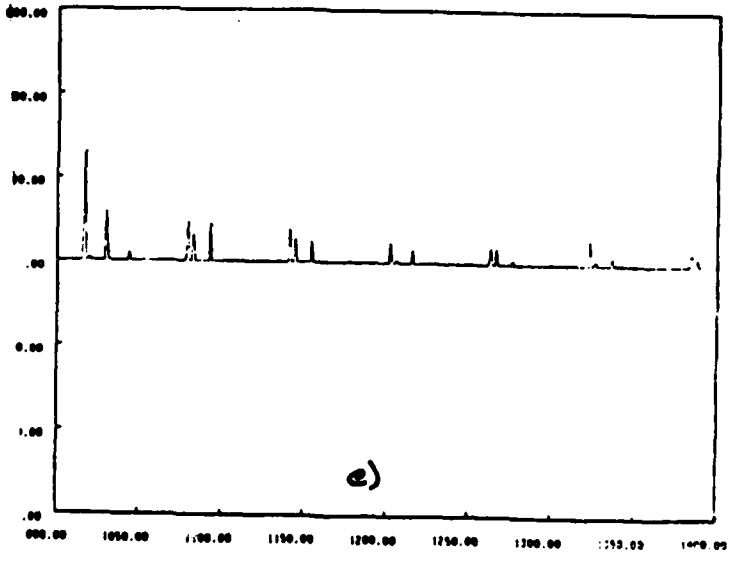
Reflection coefficients of speech data in File 8 ;orders 1- 5



Residuals of speech data in File 8 ;orders 5- 5



Gamma of speech data in File 8 ;orders 5- 5



Spectrogram of Residuals of speech data in File 8 ;orders 5- 5
(Pos. Freq. only)

FIG 14

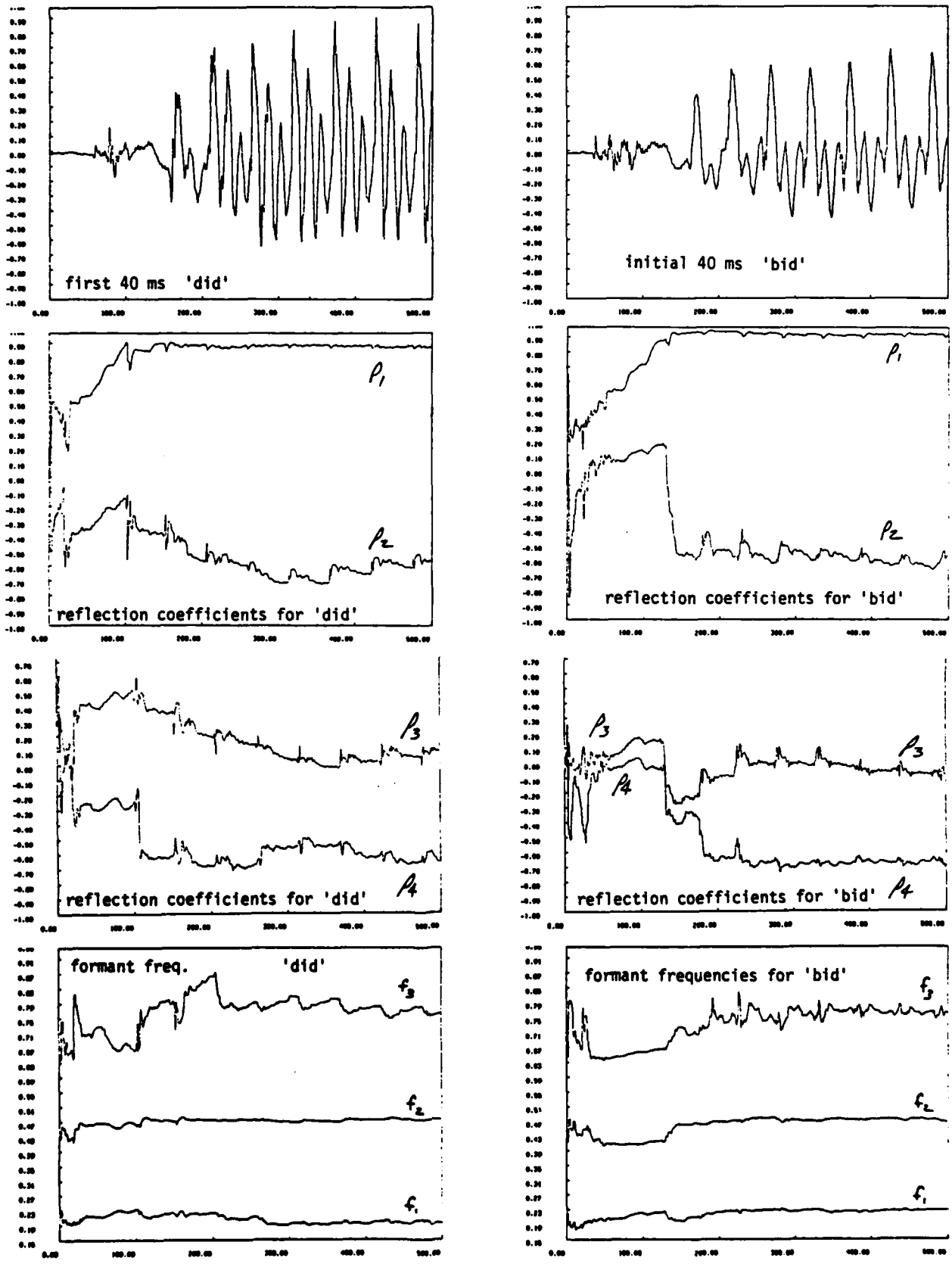
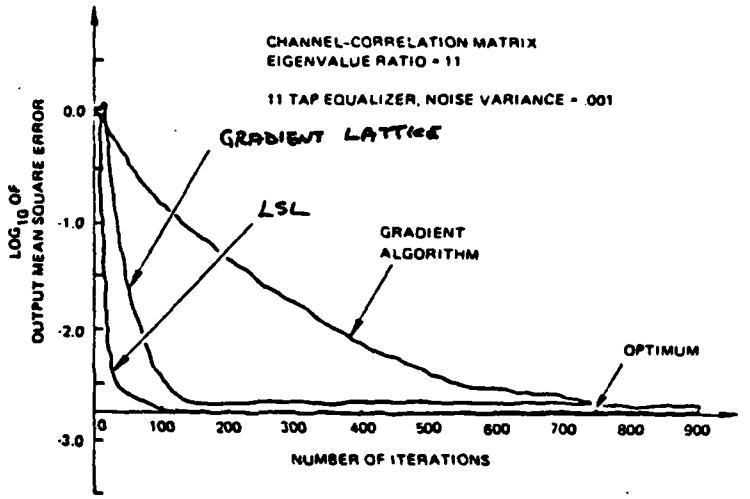


FIG 15

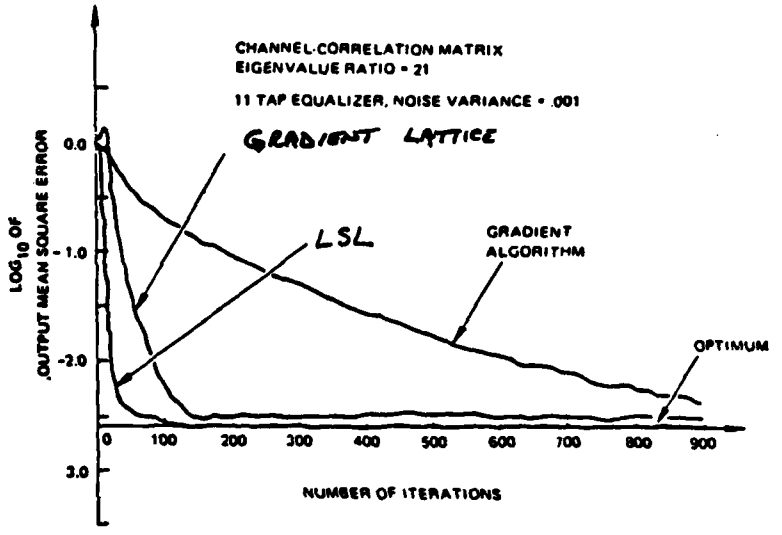
SQNLSL with $\lambda = .98$

APPLICATION FOR CHANNEL EQUALIZATION

The adaptive lattice filter offers substantial advantages for channel equalization where the orthogonalizing properties and fast tracking characteristics are important. Tapped delay line adaptive gradient equalizers, although simple to implement, have a rate of convergence that depends on the ratio of the largest to smallest eigenvalues of the channel correlation matrix [Gitlin et al, 1973]. Self-orthogonalizing techniques have been proposed by [Gitlin and Magee, 1977] and in lattice form by [Griffiths, 1977, Griffiths and Medaugh, 1979(2)]. The gradient lattice equalizer [Satorius and Alexander, 1979(2)] and the LSL equalizer (see Section 6) [Satorius and Pack, 1981] have been shown to provide very fast convergence. The lattice filter equalizers demonstrated fast convergence that was independent of the channels eigenvalues disparity ratios, see Fig. 16 from [Satorius and Pack, 1981]. Two simulated data channels with correlation matrices of eigenvalues disparity ratios (ratio of largest to smallest eigenvalues) of 11 and 21 respectively were studied. An 11 tap equalizer was implemented using the LMS gradient algorithm, the gradient lattice algorithm and the LSL algorithm. The gradient tapped delay line equalizer had considerably slower convergence that depended on the eigenvalue ratio. The LSL equalizer converged in both cases in approximately 40 iterations while the adaptive lattice equalizer required approximately 120 iterations to converge.



Comparison by simulation of convergence properties for eigenvalue ratio = 11.

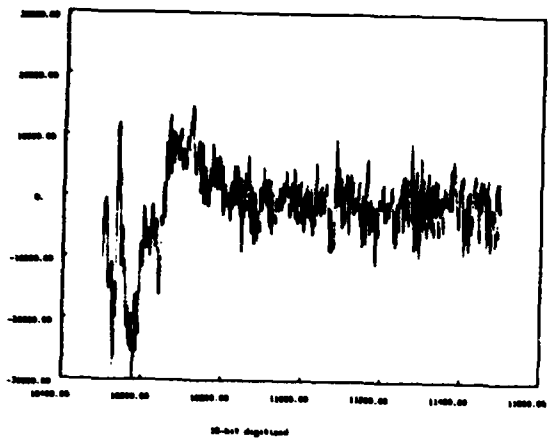


Comparison by simulation of convergence properties for eigenvalue ratio = 21.

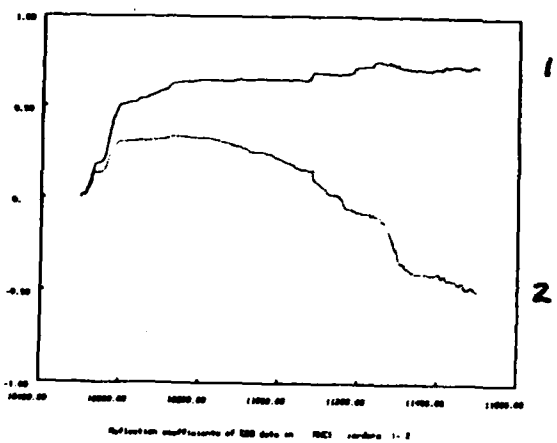
Figure 16

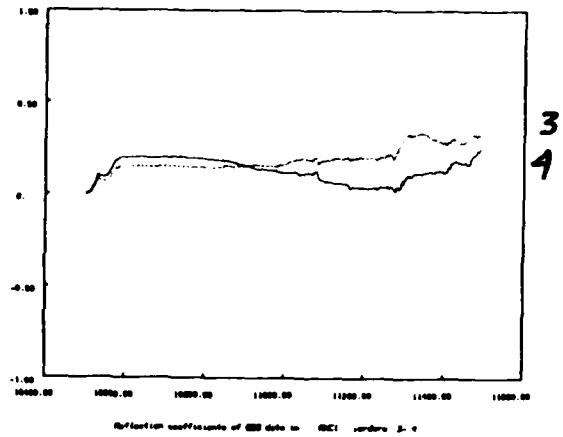
APPLICATION FOR ELECTROENCEPHALOGRAPHIC ANALYSIS

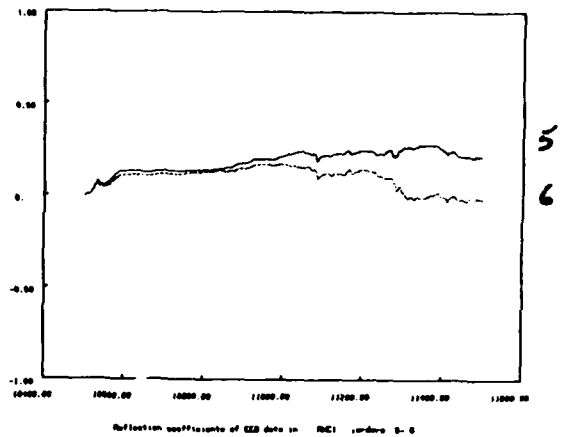
Electroencephalographic (EEG) data analyzed by autoregressive modeling can provides a better summary of EEG spectral information than frequency domain techniques, such as FFTs. The reflection coefficients from the SQNLSL algorithm were studies to detect subtle changes in brain states as observed in EEG activity [Redington and Turner]. The data obtained from the left central EEG (C1) response of an adult human subject monitored during sleep onset (sampled 60 times a second) is shown in Fig. 17a. A large change in activity appears near the beginning of the raw EEG data trace and is apparent in the reflection coefficients, Fig. 17b,c,d. A second change in activity near the end of the trace is barely noticeable in the raw data; yet, it is easily recognized in the activity of the higher order reflection coefficient. The changes in reflection coefficients may reflect physiological transitions and provide a means of inferring presence or sequence of EEG brain states.



EEG
CENTRAL
C1 ELECTRODE







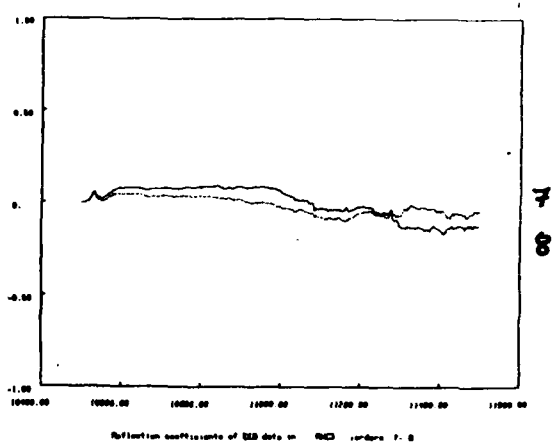


FIG 17

10. COMMENTS AND CONCLUSIONS

The lattice filter structure as a realization of a digital transfer function has several advantages; it is a cascade of identical sections, has a general insensitivity to round-off noise, and the reflection coefficients can be related to physical processes. The physical interpretation of reflection coefficients gives them intuitive appeal, particularly for speech signals. For adaptive estimation, the lattice structure is the natural form for an efficient solution to recursive least squares problems. Lattice filters provide an orthogonalization or decoupling of the states of the input process. The stability of an all pole model when expressed in lattice form can be determined by inspection.

The real advantage of the lattice structure lies in adaptive estimation and filtering. An N stage lattice filter automatically generates all the outputs which would be generated by N different TDL filters with lengths from 1 to N. This allows dynamic assignment of any filter length which proves most effective at any instant of adaptive processing. When compared to the simpler adaptive transversal filter, the lattice filter has superior convergence properties and reduced sensitivity to finite wordlength effects.

Recursive lattice estimation algorithms allow the exact least squares solution to be efficiently updated for each new time sample. The structure of this exact recursive approach is similar to the gradient lattice techniques; however an optimal gain is calculated at every time sample. This optimal recursive solution has a complexity that is only slightly more that the gradient lattice solution. Consequently, the LSL and SQNLSL algorithms achieve extremely fast initial convergence and can track quickly time varying parameters. The SQNLSL has a very compact notation and normalizes all signals to unit variance at each stage. A single integrated circuit to execute this algorithm has been proposed.

However, as with all adaptive estimation procedures, there are various trade-offs to be made. The lattice structure involves more computation and is conceptually more complicated than the tapped delay line structure but has better convergence properties. The recursive least squares lat-

tice offers even better convergence than the gradient lattice, but again it is slightly more complex. For example, in the definition of the two reflection coefficients, there is a difference in the time subscripts of the normalizing covariances. In the stationary case, these terms are identical but in the LSL the difference is critical; in general the algorithm will fail if this difference is overlooked [Satorius and Shensa, 1980]. The SQNLSL allows a very short time constant to be applied to the sampled data so that the quickly time varying nature of the signal can be tracked. However, attempting to track transient speech sounds also tracks the pitch excitation signal which was not of interest. For processes that tend toward stationarity, the convergence properties of the gradient lattice and LSL lattice are similar [Honig, 1983].

Many extensions to the basic recursive least squares algorithm have been developed. Reviews of least squares adaptive lattice filtering can be found in [Satorius and Shensa, 1980, Friedlander, 1982(3)]. Recursive ladder algorithms for ARMA modeling have been presented in [Lee et al, 1982]. The SQNLSL algorithm has been extended from the "pre-windowed" data case presented here to the "Covariance" data case in [Porat et al, 1982]. The problem of system identification has been addressed in [Porat and Kailath, 1983]. A review of lattice filters for nonstationary processes was presented in [Kailath, 1982].

There are other means to implement the lattice filter structure for estimation. The order-update recursions can also be obtained by using a Cholesky decomposition of the covariance matrix [Dickinson, 1979(1), Dickinson and Turner, 1979(2), Klein and Dickinson]. Alternatively, since a reflection coefficient is similar to a correlation coefficient, computationally simple techniques to estimate correlation coefficients can be applied to determining the reflection coefficients. Since the correlation of Gaussian random variables is related to the correlation of the hardlimited variables by an ARCSIN relationship, a very simple reflection coefficient approximation technique is possible [Turner et al, 1980]. This algorithm requires only a count of polarity changes in the prediction errors to estimate the reflection coefficients (assuming zero mean unit variance Gaussian signals).

Overall the adaptive lattice filter offers a compact algorithm for obtaining quickly converg-

ing estimates. The properties of the lattice filter and reflection coefficients motivate their use in many practical situations.

ACKNOWLEDGEMENT

The author would like to acknowledge the encouragement, assistance and advice of B. Dickinson and M. Morf on portions of this manuscript.

11. REFERENCES

Ahmed, 1981(1) — Ahmed, H.M., "VLSI Architectures For Real-Time Signal Processing," Ph.D. Dissertation, Electrical Engineering, Stanford University, Stanford, CA, 1981.

Ahmed et al, 1981(2) — Ahmed, H.M., M. Morf, D.T. Lee and P.H. Ang, "A Ladder Filter Speech Analysis Chip Set Utilizing Co-ordinate Rotation Arithmetic," *Proc. Int. Symp. on Circuits and Systems,* Chicago, IL, April 1981.

Ahmed et al, 1981(3) — H.M. Ahmed, M. Morf, D.T. Lee and P.H. Ang, "A VLSI Speech Analysis Chip Set Based on Square-Root Normalized Ladder Forms," *IEEE Int. Conf. Acoustics, Speech and Signal Processing,* March 1981, pp. 648-653.

Ahmed et al, 1982 — Ahmed, H.M., J.-M. Delosme and M. Morf, "Highly Concurrent Computing Structures for Digital Signal Processing and Matrix Arithmetic," *IEEE Computer Magazine,* Feb. 1982, pp. 65-82.

Barndorff-Nielsen and Schou — Barndorff-Nielsen, O and G. Schou, "On the Parameterization of Autoregressive Models by Partial Autocorrelation," *J. Multivariate Anal.* Vol. 3, pp. 408-419, 1973.

Burg, 1967 — Burg, J.P., "Maximum Entropy Spectral Analysis," paper presented at the 37th Ann. Int. Meeting, Soc. of Explor. Geophys., Oklahoma City, OK, October 31, 1967.

Burg, 1975 — Burg, J.P., "Maximum Entropy Spectral Analysis," Ph.D. Dissertation, Stanford University, Stanford, CA, 1975.

Chu and Messerschmitt, 1980 — Chu,P.L. and D.G. Messerschmitt, "Zero Sensitivity Analysis of the Digital Lattice Filter", *IEEE Int. Conf. Acoustics, Speech and Signal Processing,* April 1980, pp. 89-93.

Chu and Messerschmitt, 1983 — Chu,P.L. and D.G. Messerschmitt, "Zero Sensitivity Properties of the Digital Lattice Filter", *IEEE Trans. Acoustics, Speech and Signal Processing,* Vol. ASSP-31, No. 3, June 1983, pp. 685-706.

Dickinson, 1979(1) — Dickinson, B.W., "Estimations of Partial Correlation Matrices Using Cholesky Decomposition," *IEEE Trans. Automatic Control,* Vol. AC-24, No. 2, April 1979, pp. 302-305.

Dickinson and Turner, 1979(2) — Dickinson, B. and J. Turner, "Reflection Coefficient Estimation Using Cholesky Decomposition" *IEEE Trans. Acoustics, Speech and Signal Processing,* ASSP-27, No. 2, April 1979, pp. 146-149.

Flanagan — **Flanagan, J. L., *Speech Analysis Synthesis and Perception,* Springer-Verlag, New York, 1972.**

Friedlander, 1982(3) — Friedlander, B., "Lattice Filters for Adaptive Processing", *Proceedings of the IEEE,* Vol. 70, No. 8, August 1982, pp. 829-867.

Gibson and Haykin — Gibson, C.J. and S. Haykin, "Learning Characteristics of Adaptive Lattice Filtering Algorithms", *IEEE Trans. Acoustics, Speech, and Signal Processing,* Vol. ASSP-28, No. 6, Dec. 1980, pp. 681-691.

Gitlin et al, 1973 — Gitlin, R.D., J.E. Mazo and M.G. Taylor, "On the Design of Gradient Algorithms for Digitally Implemented Adjustment Filters", *IEEE Trans. on Circuit Theory*, Vol. CT-20, March 1973, pp. 125-136.

Gitlin and Magee, 1977 — Gitlin, R.D. and F.R. Magee, "Self-Orthogonalizing Adaptive Equalizer Algorithms", *IEEE Trans. Communications*, Vol. COMM-25, No.7, July 1977.

Gopinath and Sondhi, 1970 — Gopinath, B. and M.M. Sondhi, "Determination of the Shape of the Human Vocal Tract from Acoustical Measurements", *Bell Systems Technical Journal*, Vol. 49, pp. 1195-1214, 1970.

Gopinath and Sondhi, 1971 — Gopinath, B. and M.M. Sondhi, "Inversion of the Telegraph Equation and the Synthesis of Nonuniform Lines", *Proceedings of the IEEE*, Vol. 59, No. 3, 1971, pp. 383-392.

Gray and Markel, 1973 — Gray, A.H., Jr. and J.D. Markel, "Digital Lattice and Ladder Filter Synthesis," *IEEE Trans. Audio Electroacoustics*, Vol. AU-21, No. 6, pp. 491-500, 1973.

Gray and Markel, 1975 — Gray, A.H., Jr. and J.D. Markel, "A Normalized Digital Filter Structure," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, No. 3, pp. 268-277, 1975.

Griffiths, 1977 — Griffiths, L.J., "A Continuously Adaptive Filter Implemented As A Lattice Structure," *IEEE Int. Conf. Acoustics, Speech, and Signal Proc.*, May 1977, pp. 683-686.

Griffiths, 1978 — Griffiths, L.J., "An Adaptive Lattice Structure for Noise-Cancelling Applications," *IEEE Int. Conf. Acoustics, Speech, and Signal Proc.*, April 1978, pp. 87-90.

Griffiths, 1979(2) — Griffiths, L.J., "Adaptive Structures for Multiple-Input Noise Cancelling Applications," *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, April 1979, pp. 925-928.

Griffiths and Medaugh, 1979(2) — Griffiths, L. and R.S. Medaugh, "Convergence Properties of An Adaptive Noise Cancelling Lattice Structure," *Proc. 1978 IEEE Conf. on Decision and Control*, San Diego, CA, pp. 1357-1361.

Grenander and Szego — Grenander, U. and G. Szego, *Toeplitz Forms and Their Applications*, University of California Press, Berkeley, CA, 1958.

Honig, 1983 — Honig, M.L., " Convergence Models for Lattice Joint Process Estimators and Least Squares Algorithms", *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-31, No. 2, April 1983, pp. 415-425.

Honig and Messerschmitt, 1981 — Honig, M.L. and D.G. Messerschmitt, " Convergence Properties of an Adaptive Digital Lattice Filter", *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, No. 3, June 1981, pp. 642-653.

Horvath — Horvath, S. "Lattice form adaptive recursive digital filters: Algorithms and Applications", *Proc. 1980 Conf. Circuits Systems*, (Houston, Texas), pp. 128-133.

Itakura and Saito, 1968

Itakura, F. and S. Saito, "Analysis Synthesis Telephony Based upon the Maximum Likelihood Method," *Rep. 6th International Congress Acoust.*, Y. Konasi, Ed., Tokyo, Japan, Rep. C-5-5, Aug. 21-28, 1968.

Kailath, 1982 — Kailath, T., "Time-variant and Time-invariant Lattice Filters for Nonstationary Processes", *Mathematical Tools and Models for Control Systems Analysis and Signal Processing,* CNRS Editions, Vol. 2, Paris, pp. 417-464, 1982.

Kelly and Lochbaum — Kelly, J. L. Jr. and C. C. Lochbaum, "Speech Synthesis", *Proc. 4th International Congress Acoust.,* Paper G42, pp. 1-4, 1962.

Klein and Dickinson — Klein, J.D. and B.W. Dickinson, "A Normalized Ladder Form of the Residual Energy Ratio Algorithm for PARCOR Estimation Via Projections", *IEEE Trans. Automatic Control,* Vol. AC-28, No. 10, Oct. 1983, pp. 943-952.

Lee, 1980 — Lee, D.T.L., "Canonical Ladder Form Realizations and Fast Estimation Algorithms," Ph.D. Dissertation, Department of Electrical Engineering, Stanford University, Stanford, CA., August 1980.

Lee and Morf, 1980 — Lee, D.T.L. and M. Morf, " A Novel Innovation Based Approach to Pitch Detection," *IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* April 1980, pp. 40-44.

Lee et al, 1981 — Lee, D.T.L., M. Morf and B. Friedlander, "Recursive Square-Root Ladder Estimation Algorithms," *IEEE Trans. Acoustics, Speech, and Signal Processing,* Vol. ASSP-29, No. 3, June 1981, pp. 627-641.

Lee et al, 1982 — Lee, D.T.L., B. Friedlander and M. Morf, "Recursive Ladder Algorithms for ARMA Modeling," *IEEE Trans. Automatic Control,* Aug. 1982, pp. 753-764.

LeRoux and Gueguen — Le Roux, J. and C. Gueguen, "A Fixed Point Computation of Partial Correlation Coefficients," *IEEE Trans. Acoustics, Speech, and Signal Processing,* Vol. ASSP-25, No. 3, June 1977, pp. 257-259.

Makhoul, 1975 — Makhoul, J., "Linear Prediction: A Tutorial Review," *Proc. IEEE,* Vol. 63, pp. 561-580, 1975.

Makhoul, 1978(2) — Makhoul, J., "A Class of All-Zero Lattice Digital Filters: Properties and Applications," *IEEE Trans. Acoustics, Speech, and Signal Processing,* Vol. ASSP-26, No. 4, Aug. 1978, pp. 304-314.

Makhoul and Viswanathan, 1978(1)

Makhoul, J. and R. Viswanathan, "Adaptive Lattice Methods for Linear Prediction", *IEEE Int. Conf. Acoustics, Speech, and Signal Proc.,* April 1978, pp. 83-86.

Markel and Gray, 1975(1)

Markel, J.D. and A.H. Gray, Jr., "Roundoff Noise Characteristics of a Class of Orthogonal Polynomial Structures," *IEEE Trans. Acoustics, Speech, and Signal Processing,* Vol. ASSP-23, pp. 473-486, 1975.

Markel and Gray, 1975(2)

Markel, J. D. and A. H. Gray, Jr., "Fixed-Point Implementation Algorithms for a Class of Orthogonal Polynomial Filter Structures," *IEEE Trans. Acoustics, Speech, and Signal Processing,* Vol. ASSP-23, pp. 486-494, 1975.

Markel and Gray, 1976

Markel, J. D. and A. H. Gray, Jr., *Linear Prediction of Speech*, Springer-Verlag, New York, 1976.

Mitra et al, 1977 — Mitra, S.K., P.S. Kamat and D.C. Huey, "Cascaded Lattice Realization of Digital Filters", *Circuit Theory and Applications*, Vol. 5, 1977, pp. 3-11.

Morf, 1974 — Morf, M., "Fast Algorithms for Multivariable Systems," Ph.D. Dissertation, Department of Electrical Engineering, Stanford University, Stanford, CA, 1974.

Morf et al, 1977 — Morf, M., D.T. Lee, J.R. Nickolls and A. Vieira, "A Classification of Algorithms for ARMA Models and Ladder Realizations," *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, April 1977, pp. 13-19.

Murthy and Narasimham

Murthy, V.K. and G.V.L. Narasimham, "On the Asymptotic Normality and Independence of the Sample Partial Autocorrelations for an Autoregressive Process," *Appl. Math. and Comput.*, Vol. 5, pp. 281-295, 1979.

Porat et al, 1982 — Porat, B., B. Friedlander and M. Morf, "Square Root Covariance Ladder Algorithms," *IEEE Trans. Automatic Control*, Aug. 1982, pp. 813-829.

Porat and Kailath, 1983

Porat, B. and T. Kailath, "Normalized Lattice Algorithms for Least-Squares FIR System Identification," *IEEE Trans. Acoustics, Speech and Signal Processing*, Feb. 1983, pp. 122-128.

Rabiner and Schafer, 1978

Rabiner, L.R. and R.W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, New Jersey.

Ramsey — Ramsey, F.L., "Characterization of the Partial Autocorrelation Function," *Ann. Stat.*, Vol. 2, No. 6, pp. 1296-1301.

Redington and Turner

Redington, D. and J. Turner, "Recursive Ladder Autoregressive Modeling of Electrophysiological Data", to be published in *Electroencephalography and Clinical Neurophysiology.*

Samson and Reddy — Samson, C. and V.U. Reddy, "Fixed Point Error Analysis of the Normalized Ladder Algorithm", *IEEE Trans. Acoustics, Speech and Signal Processing*, Oct. 1983, pp. 1177-1191.

Satorius et al, 1979(1)

Satorius, E.H., J.D. Smith and P.M. Reeves, "Adaptive Noise Cancelling Of A Sinusoidal Interference Using A Lattice Structure," *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, April 1979, pp. 929-932.

Satorius and Alexander, 1979(2)

Satorius, E.H. and S.T. Alexander, "Channel Equalization Using Adaptive Lattice Algorithms," *IEEE Trans. Communications*, Vol. COM-27, No. 6, June 1979, pp. 899-905.

Satorius and Pack, 1981

Satorius, E.H. and J. Pack, "Application of Least Squares Lattice Algorithm to Adaptive Equalization," *IEEE Trans. on Communications*, Vol. COM-29, No. 2, February 1981, pp. 136-142.

Satorius and Shensa, 1980

Satorius E.H. and M.J. Shensa, "Recursive Lattice Filters - A Brief Overview," *Proc. 19th IEEE Conf. Decision & Control*, Albuquerque, NM, Dec. 1980, pp. 955-959.

Shensa, 1981 — Shensa, M.J., "Recursive Least Squares Lattice Algorithms - A Geometrical Approach," *IEEE Trans. Automatic Control*, Vol. AC-26, No. 3, June 1981, pp. 695-702.

Treitel and Robinson — Treitel, S. and E.A. Robinson, "Seismic Wave Propagation in Layered Media in Terms of Communication Theory," *Geophysics*, Vol. 31, pp. 17-32, 1966.

Turner et al, 1980 — Turner, J., B. Dickinson, and D. Lai, "Characteristics of Reflection Coefficient Estimates Based on a Markov Chain Model", *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, April 1980, pp. 131-134.

Turner, 1982 — Turner,J. "Application of Recursive Exact Least Square Ladder Estimation Algorithm for Speech Recognition", *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May 1982, pp. 543-555.

Volder — Volder, J.E., "The CORDIC Trigonometric Computing Technique", *IRE Trans. on Electronic Computers*, Vol. EC-8, No. 3, Sept. 1959, pp. 330-334.

Walther — Walther, J.S., "A Unified Algorithm for Elementary Functions," *Proc. 1971 Joint Spring Computer Conf.*, July 1971, pp. 379-385.

END

FILMED

3-84

DTIC